Table of Contents

Appendix H Unsafe Train Tracks Problem Detailed Implementation Case Study

H.1 Abstract

Chapter 6 presents a case study on the Unsafe Train Tracks Problem, utilising the AIC Systems Approach to systematically address uncertainty, risk, and autonomous safety challenges in railway environments. The study applies a predictive thinking framework to model and mitigate operational hazards, particularly in autonomous perception systems. The analysis is structured into six stages, starting with uncertainty problem articulation and operational environment modelling, defining complexity, resolving complications, and forecasting emergent behaviours.
It progresses to architectural intent definition, aligning autonomous system needs with safety objectives, followed by a HazTOPS analysis to derive structured safety requirements. The study then examines Disordered AIC-Driven Black Swan Scenarios, applying ArcMatrix analysis, perspective shifts, and emergent scenario predictions to model high-risk events. Machine learning-based perception training and validation (CuneiForm Strategy Development) further refine AI models for track safety monitoring. Finally, dataset generation and validation experiments compare Black Swan-enhanced and Black Swan-lacked models, assessing AI resilience under unpredictable conditions. Key findings emphasise the necessity of robust AI safety strategies and propose recommendations for enhancing railway perception systems.

H.2 Introduction

Covered in Chapter 6, section 6.1.

H.3 A Thought Experiment

Covered in Chapter 6, section 6.2.

H.4 Stage 1: Uncertainty Problem Articulation and Operational Environment Modelling[1]

H.4.1 Predictive Thinking Pipeline 1: Appreciate the Complex of the Problem Complexity Field

In this stage of the thinking process, the architect is guided to apply 4.4.8: System Primary Purpose (PrimeP) key concept. Identifying the primary purpose requires the following thoughts:

- Identifying the systems of concern as seen in the problem.
- Identifying the Supra Complexes of concern for which the problematic systems are part.
- Defining the primary purpose of Supra Complexes.

Our initial goal is to alter the observed complexity dynamics so that the train track zone capability can effectively impact the situation. This process's output is summarised in the *Architect Prediction 1.5* H.4.1.5 subsection.

The assurance problem (which we resolve through this process) is how we know that the architect accurately predicted and comprehensively addressed the issue and discovered problem-level Black Swans. What reasoning steps did the architect follow, and what assumptions were made to arrive at this initial articulation?

Why is this critical?

Because the rest of the design, including the solution choice, will be based on the accuracy of this thought process, if we get it wrong, what can go wrong? The detailed ArcPM serves as evidence of comprehensive problem articulation, which corresponds to reducing architectural epistemic uncertainty.

How do we do that?

Well, we follow the following SECoT:

H.4.1.1 Step 1.1) Identify a list of unsafe or confusing behaviours

In this thought step, we focused on identifying unsafe or confusing behaviours within a specific problem domain, particularly concerning the intrusion of drones and human trespassers into a bounded train track zone while a train passes. This step utilised general rules to define unsafe behaviour as a type of confusing behaviour, highlighting that such actions could lead to undesirable outcomes affecting safety. We posed a predictive question aimed at uncovering unsafe behaviours that disrupt the expected functioning of the environment, ultimately leading to potential risks.

[1] See also section 6.2

The objective was to articulate a problematic situation where behaviours contradicted the intended purpose of the train track zone, identified through the lens of an architect's sphere of concern. The outcome of this step included a detailed assertion listing specific confusing or unsafe behaviours, such as drones hovering without a clear purpose, flying at low altitudes near active tracks, ineffective fencing, and disrupting surveillance systems. Each identified behaviour was described along with its unsafe aspects and undesirable outcomes, emphasising how these actions contribute to a confusing complex that undermines the safety and security of the train track zone.:

Table H.1 Identification and analysis of unsafe problematic behaviours in train track zones

SECoT Title	Unsafe problematic behaviours list identification
Input	An anonymous Drone and Human trespassers intrude into a bounded train track zone while the train passes.
General Systems Rules	**General rule A**: Unsafe behaviour is a type of Confusing Behaviour. Confusing behaviour of some situation A, in any complexity, is a behaviour that seems contradictory to what is meant to be manifested. **General rule B:** Confusing behaviours lead to undesirable emergent outcomes about some elements (of the observed complexity) with respect to some element B, whereby situation A falls within element B's sphere of concern for others.
Predictive Thinking Process	**Predictive Question 1.1.1:** What unsafe behaviours within the problem domain's Confusing Complex disrupt an expected manifestation of some purpose, potentially leading to undesired consequences, which fall into our sphere of concern? **Guiding Prompt 1.1.2:** Identify an aspect of the problem of a single informative concern which you deem to be unsafe or confusing as it does not follow an intended purpose or seems to break some desired harmony in a system. We defined the term "confusing" to include scenarios where the behaviour may not be "unsafe", for example: *A drone is observed hovering near a restricted train track zone for an extended period. Still, it does not approach any critical infrastructure or demonstrate hostile behaviour, such as approaching a moving train or interfering with*

	surveillance cameras. Instead, it stays at a fixed altitude, maintaining a distance from the tracks and infrastructure while intermittently adjusting its position within a defined perimeter. In other words, an informative concern is a problematic situation in which some system of interest is undesirably behaving in a way that contradicts the expected PrimeP. This step is considered competitive when a problematic aspect, including its causal impact, is identified. **Step completion criteria1.1.3:** The step is regarded as complete when a problem situation is described, including undesirable impact.
Output Prediction	**Architect assertion 1.1.4:** The architect asserts that: **Confusing or Unsafe Behaviours Identified:** 1. **Drone Hovering without Clear Purpose:** ○ **Description:** A drone can be observed hovering within the restricted track zone near the train but does not engage in overtly hostile actions like moving directly toward the train or surveillance equipment. ○ **Unsafe Aspect:** This behaviour contradicts the expected use of the track zone as a protected area, introducing uncertainty about whether the drone's presence is benign or potentially adversarial. ○ **Undesirable Outcome:** This ambiguous hovering distracts security attention and could normalise such intrusions, reducing vigilance and risking slow response to real threats. 2. **Drone Moving at Low Altitude near Active Tracks:** ○ **Description:** The drone flies at low altitude parallel to the track, near the train as it passes, remaining within proximity but not directly intersecting the train's path. ○ **Confusing Aspect:** A drone's presence so close to an active train suggests potential interference but does not immediately obstruct the train's passage, creating ambiguity over intent. ○ **Undesirable Outcome:** This creates a high-risk scenario if the drone's trajectory shifts unexpectedly,

	possibly leading to an accident or signalling failure, making train operators and security personnel uncertain about response protocol. 3. **Fence Ineffectiveness and Perceived Openness to Drones:** o **Description:** Despite fencing around the track zone, drones are observed freely entering the area without detection or interception, exploiting the "open air" aspect above the physical barriers. o **Confusing Aspect:** The fence suggests restricted access, yet it does not extend to aerial barriers, allowing drones to intrude without physical deterrents. o **Undesirable Outcome:** This gap creates a false sense of security, as drones can consistently access restricted areas, undermining the zone's purpose of deterring unauthorised access. 4. **Drone Disruption of Surveillance Systems without Direct Contact:** o **Description:** The drone's movement pattern occasionally disrupts surveillance feeds due to signal interference or unexpected camera angles needed to track its motion, even though the drone itself doesn't physically tamper with equipment. o **Confusing Aspect:** This behaviour interferes with track zone monitoring without direct sabotage, contradicting the expected reliable operation of surveillance. **Undesirable Outcome:** This undermines the ability of security systems to maintain continuous observation of the zone, potentially allowing other intrusions or adverse events to go undetected.

Implementation: The train track zone is not behaving as expected, which is to prevent intrusion. Instead, the zone permits drones to intrude, and people can do so easily without practical denial. The open space of the train track zone appears inviting for drones' unconstrained intrusion. Also, the fence is ineffective in enticing deterrence over human intruders. However, In this case study, we will focus on drone intrusion. Since drone intrusion is more concerning than people intrusion, the available resources and capabilities are predominantly aligned with dealing with drones, not human trespassers.

Note that the implementation has made it explicit that it will not address human trespassers and explained why the architect intends not to do so. The assertion also assumed that the intruder drone has an adversarial intent, thus directing the design process to consider not a random drone risk but rather potential adversarial intent. As you can imagine, if assumptions are not made explicit at this stage and thus validated among stakeholders, the design will run the risk of being directed towards solving unrealistic problems or wrong problems.

Here, a quality or design assurance authority gets a glimpse of why the design excludes certain aspects. This would allow us to scrutinise the design thinking right from the initial stages, thus enabling more transparent thought processes that lead to decisions. At this stage, soft aspects such as ethics can be reviewed objectively and ascertained as part of the collection of evidence to support an informative safety case.

H.4.1.2 Step 1.2) Generate a descriptive image that visualises the unsafe behaviour

In this step, we focused on generating a descriptive image to visualise the unsafe behaviour of a drone hovering within a restricted train track zone. This step aimed to enhance our understanding of the situation through a visual representation that complements the written description. Starting with the architect's assertion regarding the ambiguity and risks associated with the drone's presence, we applied general systems rules to analyse how the drone's behaviour contradicts the intended use of the area, creating uncertainty about its intentions. We posed a predictive question to explore the visual aspects of this confusing behaviour and then crafted a prompt for the DALL-E tool to create a simplistic black-and-white sketch illustrating the scenario of the drone intruding while a train passes by.

While the prompt captures the essence of the behaviour, it lacks specificity regarding the interactions and purposes of the elements involved, which could lead to varied interpretations. The architect can choose the appropriate variation. This visualisation allows the design authority team and stakeholders to critically engage with the architect's assumptions, fostering discussions that could either validate or challenge their interpretations of the problem.

Table H.2 Visual representation of unsafe and confusing drone behaviours in train track zones

SECoT Title	Unsafe problematic behaviours visualisation
Input	Architect assertion 1.1.4 1. **Drone Hovering without Clear Purpose:** o **Description:** A drone can be observed hovering within the restricted track zone near the train but does not engage in

	overtly hostile actions like moving directly toward the train or surveillance equipment. ○ **Unsafe Aspect:** This behaviour contradicts the expected use of the track zone as a protected area, introducing uncertainty about whether the drone's presence is benign or potentially adversarial. **Undesirable Outcome:** This ambiguous hovering distracts security attention and could normalise such intrusions, reducing vigilance and risking slow response to real threats.
General Systems Rules	**General rule A**: Unsafe behaviour is a type of Confusing Behaviour. Confusing behaviour of some situation A, in any complexity, is a behaviour that seems contradictory to what is meant to be manifested. **General rule B:** Confusing behaviours lead to undesirable emergent outcomes about some elements (of the observed complexity) with respect to some element B, whereby situation A falls within element B's sphere of concern for others.
Predictive Thinking Process	**Predictive question 1.2.1:** How does the confusing behaviour appear visually? **Guiding prompt 1.2.2:** Graphically visualise how you perceive the problem, confusing the whole scenario encompassing the problem aspect. You may use text-to-image generation tools such as Dall-E to generate an abstract, simplistic sketch representing how you imagine the situation. You may experiment with different prompts until you find an appropriate (detailed yet realistic) representation of the problem as you picture it. **Step completion criteria 1.2.3:** The step is considered complete when a single appropriate visual representation visualises how parties to the problem are within each other's sphere of concern and how the model depicts a situation within the architect's sphere of concern.
Output Prediction	**Architect assertion 1.2.4:** The architect asserts that the following depiction model captures the confusing behaviour faithfully:

Implementation: Using the Dall-E tool and the following prompt: “Generate a very Simplistic Sketch, black and white, that depicts the following scenario: An anonymous Drone intruded into the train tracks while the train is passing by”.

The prompt captures the apparent actual behaviour but does not precisely specify the interacting elements' purposes and how the behaviour is confusing. The quality of confusion is implicit within the explicit mention of the general rule used, and the architect considers it problematic. If reused, the prompt may yield different results, but the concept remains conceptually the same.

Note that the visualisation clearly describes how the architect perceived the situation. Such an approach demonstrates the architect's commitment to transparency in the thought process involved in the design. This would allow the design authority team to question the accuracy of what the architect thinks of the problem. Furthermore, stakeholders may argue for or against the assumptions presented.

H.4.1.3 Step 1.3) Define the complex of the complexity field

In this step, we aimed to define the complex of the complexity field by examining the visual scenario of the drone hovering near the train tracks. The designer focused on identifying the interconnected elements that contribute to the problem, thus gaining insight into how these components coexist and influence one another within the context of the situation. Guided by the architect's previous assertion regarding the visualisation, we applied general systems rules to understand the complexity of the scenario, considering that any complexity consists of coexisting elements, regardless of whether they actively interact. We formulated a predictive question to ascertain the specific elements involved in the problematic situation as represented in the visual model.

Following this analysis, we created a guiding prompt to infer a complex of elements perceived to be part of the scenario. The completion of this step involved generating a list of elements identified in the depiction. The architect ultimately asserted that the following list accurately represents the complex of concerns: {train, intelligent adversarial drone, train tracks zone fence, vegetation around the fence, electric power lines}. This structured list encapsulates the critical components contributing to the observed complexity, ensuring that all relevant systems are acknowledged and that no implausible or omitted elements exist in the context of the scenario.

Table H.3 Defining the complex of complexes in problematic train track zone scenarios

SECoT Title	The complex of complexes definition
Input	**Architect assertion 1.2.4:** The architect asserts that the following depiction model captures the confusing behaviour faithfully: 
General Systems Rules	**General rule C:** Complexity is a field containing an organisational experience of a phenomenon concerning a general problem-solver (such as a Predictive Observer). It involves an operational environment of coexisting complexes and the complicated nature (or types) of their relationships and interactions (epistemic uncertainty) for a Predictive Observer to predict their past, present, and future situations (managing aleatoric uncertainty or randomness).
Predictive Thinking Process	**Predictive question 1.3.1:** What is the Complex of coexisting elements involved in the problematic situation, as observed in the model? **Guiding prompt 1.3.2:** Guided by the visual representation, infer a complex of complexes you imagined to be part of your perceived scenario. **Step completion criteria 1.3.3:** The step is considered complete when a list of elements is defined and can be identified in the visualisation model.
Output Prediction	**Architect assertion 1.3.4:** The architect asserts that: The following list represents the complex of concerns: {train, intelligent adversarial drone, train tracks zone fence, vegetation around the fence, electric power lines}

The output is considered acceptable when the output list contains all systems. It is unacceptable if the list includes elements not seen in the depiction or implausible in the specified situation's complexity. Also, there are missing elements that should be mentioned in the list.

Implementation: Examine the depiction and all content mentioned in the analysis so far and generate a list of elements.

H.4.1.4 Step 1.4) Define the supra-complexes and their PrimePs

In this step, we focused on identifying the supra-systems related to the problematic components previously analysed. The objective was to clarify the primary purpose of each identified part to understand their roles within the larger context better. Guided by the rules of the general system, we defined the Primary Purpose (PrimeP) that each component serves, which helps illuminate the overall goals necessary for each part to achieve. To facilitate this understanding, we posed a predictive question to uncover the interacting supra-complexes that encompass the coexisting elements. A guiding prompt encouraged us to view these systems as parts of larger supra-complexes, thus enabling us to define the associated groups that share a common purpose.

Upon completion of this step, we successfully generated a list of supra-complexes that included all observed elements. We established “Train Network” as one supra-complex, incorporating the following components: {Train, Train tracks zone fence, Vegetation around the fence, Electric power lines}. In recognition of the adversarial intent of the intruding drone, we introduced “Adversarial Scheme” as a second supra-complex, encompassing the {adversarial drone}. This classification implies that the adversarial drone is part of a broader, more complex scheme targeting the Train Network. The architect concluded that these two supra-complexes, {Train Network, Adversarial Scheme}, capture the components' critical interrelations and overarching concerns within the problem domain.

Table H.4 Definition of supra complexes in train network and adversarial scheme context

SECoT Title	Supra-complexes and their PrimePs definition
Input	**Architect assertion 1.3.4:** The architect asserts that: The following list represents the complex of concerns: {train, intelligent adversarial drone, train tracks zone fence, vegetation around the fence, electric power lines}
General Systems Rules	**General rule D:** Supra Complex, Supra Source and Supra Sink, AIC Systems Approach. A **Supra Complex** is a relatively larger collection of complexes where a complex of interest (of a predictive observer) is part of. **General rule E: Requisite consistent purpose:** An Ideal System is forever consistently purposeful, which means having a clear purpose with respect to the architecting observer that is clear to all possible observers and in all possible scenarios.

<table>
<tr><td></td><td>General rule F: System Primary Purpose (PrimeP), AIC Systems Approach</td></tr>
<tr><td>Predictive Thinking Process</td><td>
Predictive question 1.4.1a: What are the observed interacting Complexes, the possible holistic PrimeP they serve, and their Primary capability (or function) that define their operational situation?

Predictive question 1.4.1b: What are the interacting Supra Complexes of concern of which the coexisting elements are parts?

Predictive question 1.4.2: What is the PrimeP for every Supra Complex such that it is expected to manifest in all possible scenarios?

Guiding prompt 1.4.3: to help with answering 1.4.1a, used the following table:
<table>
<tr><th>Observed System</th><th>Primary Purpose (PrimeP)</th><th>Primary Capability (PrimeC)</th></tr>
<tr><td>Complex</td><td>Possible holistic PrimeP to serve</td><td>The primary function that delivers the PrimeP is written in the format of {adjective_noun}</td></tr>
</table>
As for 1.4.1b, consider the identified systems as parts of Supra Complexes. Then, define the associated Supra Complexes, encompassing systems with a common purpose. Supra Complexes are written in a capitalised format.

Guiding prompt 1.4.4: Infer the PrimeP for every Supra Complex. The choice of the PrimeP will guide the rest of the design on what priorities each Supra Complex intends to achieve. Getting the priorities focused only on specific aspects or generally on broader aspects dictates the design scope. For example, suppose we have a Road Transportation as the Supra Complex. Suppose we choose a PrimeP to mobilise people across a single specific strip of road. In that case, we will constrain the design decisions to be influenced by the limited, finite stretch of the operational domain.

Step completion criteria 1.4.5: The step is considered complete when a list of Supra Complexes encompasses all the observed visible elements. Also, a PrimeP is defined for each Supra Complex.
</td></tr>
<tr><td>Output Prediction</td><td>
Architect assertion 1.4.6: The architect asserts that:

There are two Supra Complexes of concern:

{Train Network, Adversarial Scheme}, whereby:

Train Network: {train, train tracks zone fence, vegetation around the fence, electric power lines}

Adversarial Scheme: {adversarial drone}
</td></tr>
</table>

	The adversarial drone is assumed to be part of a larger Adversarial Scheme targeting the Train Network. **Train Network PrimeP:** transports passengers and goods along the designated train tracks. **Adversarial Scheme PrimeP:** Disrupt Train Network operations.

See Table J.3 in Appendix J for the table required in Guiding Prompt 1.4.3. “Train Network” is a Supra Complex encompassing the following systems: {train, train tracks zone fence, vegetation around the fence, electric power lines}. Due to the adversarial intent of the intruding drone, we will assume the “Adversarial Scheme” to be a Supra Complex that encompasses {adversarial drone}. This means that more parts will be involved in the Supra Complex; however, they are invisible from the observation. Output is a list of Supra Complexes and their parts. The naming convention of the Supra Complexes should be capitalised.

According to the PrimeP definition, a train network's ultimate purpose is to transport people and goods safely and securely. Adversarial Schemes primarily disrupt train network operations.

It is possible to choose various ways to define PrimeP for Supra Complexes. For example, suppose the train network transports people and goods only. However, choosing such a definition reduces the emphasis on other critical qualities, such as “safety,” which may lead to setting the design process assuming AIC goals that only focus on getting people and goods from A to B, regardless of how safe the journey is. Hence, we include “safety” as part of the PrimeP. We also consider the safety requirements to be only within the designated train tracks and not in general.

As for the Adversarial Scheme, there can be many reasons for what PrimeP is. Suppose we choose a systemic attack on the broader national interest. In that case, the problem domain will drastically differ from localising the PrimeP to only on that specific strip of train track zone. We will introduce an additional assumption that targets the disruption of train transportation across the Train Network, positioned between the two extreme assumptions. This way, we will account for a worst-case scenario that goes beyond a single train track strip. Such a decision is to be made by the architect and the stakeholders. The job of the systems approach is to inspire the stakeholders and practitioners to ask important, deep questions so we can minimise the overall epistemic uncertainty based on the worst-case scenario principle.

H.4.1.5 Architect Prediction 1.5:

Given the above list of thinking steps, architect prediction entails the following definition of problem domain complexity:

Table H.5 Architect output prediction

Architect Prediction

Architect assertion 1.1.4: The architect asserts that:
 	
1. **Drone Hovering without Clear Purpose:**
   - **Description:** A drone can be observed hovering within the restricted track zone near the train but does not engage in overtly hostile actions like moving directly toward the train or surveillance equipment.
   - **Unsafe Aspect:** This behaviour contradicts the expected use of the track zone as a protected area, introducing uncertainty about whether the drone's presence is benign or potentially adversarial.
   - **Undesirable Outcome:** This ambiguous hovering distracts security attention and could normalise such intrusions, reducing vigilance and risking slow response to real threats.
2. **Drone Moving at Low Altitude near Active Tracks:**
   - **Description:** The drone flies at low altitude parallel to the track, near the train as it passes, remaining within proximity but not directly intersecting the train's path.
   - **Confusing Aspect:** A drone's presence so close to an active train suggests potential interference but does not immediately obstruct the train's passage, creating ambiguity over intent.
   - **Undesirable Outcome:** This creates a high-risk scenario if the drone's trajectory shifts unexpectedly, possibly leading to an accident or signalling failure, making train operators and security personnel uncertain about response protocol.
3. **Fence Ineffectiveness and Perceived Openness to Drones:**
   - **Description:** Despite fencing around the track zone, drones are observed freely entering the area without detection or interception, exploiting the "open air" aspect above the physical barriers.
   - **Confusing Aspect:** The fence suggests restricted access, yet it does not extend to aerial barriers, allowing drones to intrude without physical deterrents.
   - **Undesirable Outcome:** This gap creates a false sense of security, as drones can consistently access restricted areas, undermining the zone's purpose of deterring unauthorised access.
4. **Drone Disruption of Surveillance Systems without Direct Contact:**

- **Description:** The drone's movement pattern occasionally disrupts surveillance feeds due to signal interference or unexpected camera angles needed to track its motion, even though the drone itself doesn't physically tamper with equipment.
- **Confusing Aspect:** This behaviour interferes with track zone monitoring without direct sabotage, contradicting the expected reliable operation of surveillance.

Undesirable Outcome: This undermines the ability of security systems to maintain continuous observation of the zone, potentially allowing other intrusions or adverse events to go undetected.

Architect assertion 1.2.4: The architect asserts that; the following depiction model captures the confusing behaviour faithfully:

Architect assertion 1.3.4: The architect asserts that; The following list represents the complex of concerns: {train, intelligent adversarial drone, train tracks, train tracks zone fence, vegetation around the fence, electric power lines}

Architect assertion 1.4.4: The architect asserts that; There are two Supra Complexes of concern:

{Train Network, Adversarial Scheme}, whereby; Train Network comprises of {Train, Train tracks zone fence, Vegetation around the fence, Electric power lines}, and Adversarial Scheme comprises of {adversarial drone}.

The architect asserts that Train Network PrimeP transports passengers and goods along the designated train tracks. Adversarial Scheme PrimeP Disrupts Train Network operations.

H.4.2 Predictive Thinking Pipeline 2: Resolve the Complicatedness pattern of the observed complexity.

Predictive Thinking Pipeline 1 helped us to clearly, objectively and justifiably define the initial impression of the problem holistically. We exposed the overall structure of the problem domain complex . in this Predictive Thinking Pipeline, we will attempt to uncover the intricate interactions among the parts with each other.

H.4.2.1 Step 2.1) Problem Interaction Analysis Using Actions Matrix

List the identified systems and define briefly how each system interacts with all other systems using the Actions Matrix.

In this step, we systematically mapped out the interactions between each system component, defining how they influence or interact with one another. Using the Actions Matrix method, we identified and documented binary relationships across the elements previously identified as key parts of the problem domain, such as the train, adversarial drone, train tracks, fences, vegetation, and electric power lines. This approach allowed us to determine 30 distinct interactions encompassing these elements' overall complexity and intricate behaviours. We classified each interaction in a concise, action-based phrase to clarify the dynamic relationships and any confusing interdependencies. This process was guided by the principles of Complicatedness and lateral thinking to account for less obvious relationships, ensuring that every interaction was captured and understood within the context of the problem's complexity.

Table H.6 Interaction Analysis of Problematic Coexisting Elements within the Train Network and Adversarial Scheme

SECoT Title	Problem Interaction Analysis
Input	**Architect assertion 1.3.4:** The architect asserts that The following list represents the complex of concerns: {train, intelligent adversarial drone, train tracks, train tracks zone fence, vegetation around the fence, electric power lines} **Architect assertion 1.4.4:** The architect asserts that There are two Supra Complexes of concern: {Train Network, Adversarial Scheme}, whereby; Train Network comprises of {Train, Train tracks zone fence, Vegetation around the fence, Electric power lines}, and Adversarial Scheme comprises of {adversarial drone}. **Architect assertion 1.5.4:** The architect asserts that; Train Network PrimeP: transports passengers and goods along the designated train tracks. Adversarial Scheme PrimeP: Disrupt Train Network operations.
General Systems Rules	**General rule F:** Complicatedness definition. **Complicatedness:** the predictability of a given observation by the predictive observer's approach to minimising their epistemic uncertainty. It is the Impact of complexes' behaviours (events in observation) on some predictive observers' confidence in making decisions to manage, use or interact with the observed complexity field. **General rule G:** Actions Matrix.

Predictive Thinking Process	**Predictive question 2.1.1:** What is the complete set of interactions among the defined complex within problem domain complexity? **Guiding prompt 2.1.2:** Apply the Actions Matrix method and briefly describe the interactions among the list of contributing problematic coexisting elements as uncovered by Predictive Thinking Pipeline 1. Describe the interaction between the source and sink in a single verbal phrase. For every interaction where the interacting elements do not make sense, utilise the Lateral Predictive Thinking Process defined in section 5.3, and include at the end of the interaction the auxiliary third-party element where the intricate interactions made sense. Define the interaction using the following format : [source system][action][sink system]. **Step completion criteria 2.1.3:** The step is considered complete when; All binary relationships have been identified among the identified list of problematic coexisting elements. Interactions can be defined using the following format : [source system][action][sink system] (optional format).
Output Prediction	**Architect assertion:** The architect asserts that: The following list defines the complicatedness of the problem domain complexity: n1: adversarial drone approaches the train n2: train tracks guide and stabilise the train n3: track zone fence isolates train n4: vegetation obstructs the passage of the train n5: electric powerlines electricity-supplies train n6: train approaches adversarial drone n7: train tracks guide roaming adversarial drone n8: track zone fence permits adversarial drone n9: vegetation visually complicates adversarial drone [Lateral perspective with respect to an autonomous system] n10: electric powerlines physically obstruct adversarial drone n11: train drives over train tracks n12: adversarial drone follows train tracks n13: track zone fence isolates train tracks n14: vegetation around the fence visually complicates train tracks [Lateral perspective with respect to an autonomous system]

	n15: electric powerlines visually complicate train tracks [Lateral perspective with respect to an autonomous system]
	n16: train maintains course along track zone fence
	n17: adversarial drone crosses track zone fence
	n18: train tracks visually complicate track zone fence [Lateral perspective with respect to an autonomous system]
	n19: vegetation around the fence visually complicates track zone fence [Lateral perspective with respect to an autonomous system]
	n20: electric powerlines visually complicate track zone fence [Lateral perspective with respect to an autonomous system]
	n21: Train visually agitates vegetation around the fence [Lateral perspective with respect to an autonomous system]
	n22: adversarial drone avoids vegetation around the fence
	n23: train tracks visually complicate vegetation around the fence [Lateral perspective with respect to an autonomous system]
	n24: track zone fence visually complicates vegetation around the fence [Lateral perspective with respect to an autonomous system]
	n25: electric powerlines visually complicate vegetation around the fence [Lateral perspective with respect to an autonomous system]
	n26: train sparks and wobble electric powerlines wires [Lateral perspective with respect to train pantograph connection]
	n27: adversarial drone avoids electric powerlines
	n28: train tracks visually complicate electric powerlines [Lateral perspective with respect to an autonomous system]
	n29: track zone fence visually complicates electric powerlines [Lateral perspective with respect to an autonomous system]
	n30: vegetation around the fence visually complicates electric powerlines [Lateral perspective with respect to an autonomous system]
	n31: Police Officer Support Train
	n32: Police Officer capture adversarial drone
	n33: Police Officer monitor train tracks
	n34: Police Officer monitor track zone fence
	n35: Police Officer Ignore vegetation around fence
	n36: Police Officer Monitor electric powerlines wires

	n37: Train complicate Police Officer n38: adversarial drone distress Police Officer n39: train tracks Permit access Police Officer n40: track zone fence support Police Officer n41: vegetation around fence complicate Police Officer n42: electric powerlines wires Complicate Police Officer

The output is considered acceptable when;

1) All relationships are identified.
2) All relationships are briefly described in a single verbal phrase, no more than three words, and the nature of the problematic behaviours (actions). Only actions are defined.
3) Interactions are defined using the following format : [source system][action][sink system].

The output is considered unacceptable if:

1) There are missing interactions.
2) Interactions are defined using AIC factorisation method.
3) Interactions are not defined using the following format : [source system][action][sink system].

Implementation: Predictive Thinking Pipeline 1 defines the following list of problematic coexisting elements that are responsible for the confusion:

{train, intelligent adversarial drone, train tracks, train tracks zone fence, vegetation around the fence, electric power lines}. Those elements are categorised as parts that belong to the following source Supra Complexes: {Train Network, Adversarial Scheme}. By mapping the parts in a cross-product fashion, we identify 30 potential intricate behaviours which constitute a holistic view of the problem domain's complicatedness.

The format of the Actions Matrix would include the following interactions:

Table H.7 Actions Matrix representing interactions among problematic coexisting elements in the complexity field

	train	**intelligent adversarial drone**	**train tracks**	**train track zone fence**	**vegetation around the fence**	**electric powerlines**	**Police Officer**
train		n 6	n 11	n 16	n 21	n 26	n 37
intelligent adversarial drone	n 1		n 12	n 17	n 22	n 27	n 38
train tracks	n 2	n 7		n 18	n 23	n 28	n 39
train track zone fence	n 3	n 8	n 13		n 24	n 29	n 40

vegetation around the fence	n 4	n 9	n 14	n 19		n 30	n 41
electric powerlines wires	n 5	n 10	n 15	n 20	n 25		n 42
Police Officer	n 31	n 32	n 33	n 34	n 35	n 36	

Note that we utilised lateral thinking to solve the intricate description of the interactions, which we included a reference third-party observer [Lateral perspective with respect to ...]. To give an example of how we applied the Objective Lateral Predictive Thinking Process, we include only a :

We were unable to predict what relevant interactions would occur between the source “train” and the sink “electric powerline”. Thus, we needed to evoke a lateral thinking process to help us imagine what could happen. We eventually arrived at the following prediction:

n26: train sparks and wobble electric powerline wires [Lateral perspective concerning train pantograph connection]

We used the following thought process (implementing section 5.5: General Lateral Predictive Thinking Process):

Applying the Thought Process to Predict the Interaction: Train vs. Electric Power Line:

Using a lateral perspective, we will systematically apply the F.9 Thought Process to resolve and predict the interaction between a train (source) and an electric power line (sink). Our goal is to explore and characterise the new prediction (n26) between Train and Power line.

Step 1: Assume a Broad Perspective Without Anchoring to Existing Rules

This step challenges fixed assumptions and considers all potential contributors to emergent interactions.

Thought Step 1.1: Define Context Without Preconceptions

Rather than assuming a train pantograph connects to a powerline statically and predictably, we broaden our perspective:

- The train’s motion is not perfectly smooth; it involves minor vibrations, oscillations, and momentary force variations.
- The pantograph does not just connect to the powerline; it also wobbles, shifts, and momentarily loses contact, causing sparks and micro-instabilities.
- The powerline is not static; it is an elastic system that moves under mechanical influences, such as wind and pantograph-induced oscillations.

Thus, rather than assuming that a powerline is simply an energy supplier, we recognise it as an actively interacting complex affected by external dynamic forces.

Thought Step 1.2: Analyse Direct and Indirect Interactions

Direct (Control) Interactions:

- The train controls the pantograph, forcing it to maintain contact with the powerline.
- The pantograph exerts pressure on the powerline, inducing oscillations.
- The mechanical force from the train affects power transmission stability (e.g., momentary voltage fluctuations).

Indirect (Influence/Appreciation) Interactions:

- The sparks generated from momentary pantograph disconnections might interfere with electronic sensors, observing perception systems, or communication links near the train.
- The powerline oscillations might introduce electromagnetic interference (EMI) affecting adjacent electronic equipment. Like Eagle Drone.
- The Eagle Drone's perception system must appreciate these effects to adjust predictive models and avoid misinterpreting spark flashes as anomalies.

Thus, the traditional view of a train drawing power from a line evolves into a more complex interaction in which the powerline is an elastic, oscillatory system dynamically affected by train movement.

Step 2: Generate Analogies Across Different Contexts

This step broadens our view by drawing comparisons from unrelated domains to highlight hidden dynamics.

Thought Step 2.1: Develop Analogies

To understand how a train affecting a powerline may become a relevant issue in the future, we can draw analogies from different systems where motion-induced forces generate unexpected consequences:

1. Electric Guitar String and Finger Vibrations:
    - A guitar string does not simply stop vibrating when plucked; it continues oscillating, producing harmonics.
    - Likewise, when the pantograph loses contact and re-engages, the powerline behaves like a vibrating string, oscillating beyond what is typically considered.
2. Highway Traffic and Road Bumps:

- A vehicle driving at high speeds over uneven pavement causes momentary loss of tyre contact, similar to how a pantograph momentarily disconnects from a powerline when the train crosses track irregularities.

By mapping these analogies, we refine our understanding:

- The train-pantograph connection is dynamic, not static.
- Momentary disconnects lead to sparking, and energy oscillations affect the powerline system.
- These effects are not just nuisances but potentially safety-critical if ignored.

Thought Step 2.2: Explore Counterintuitive Scenarios

To stimulate unconventional thinking, we consider humorous or surprising scenarios where train-powerline interactions go beyond common expectations:

- It would be funny if… the pantograph momentarily launched sparks so large that nearby sensors mistook them for fireworks and triggered an emergency response.
- It would be funny if… powerline vibrations caused by pantograph movement accidentally signalled a nearby autonomous drone, confusing it into interpreting the oscillations appearing to be like tree branches, which may cause false positives or negatives.

These scenarios illustrate how overlooked interactions can unexpectedly affect unrelated systems, reinforcing the importance of including them in safety-critical models.

Step 3: Formulate Hypotheses and Generate New Operational Scenarios

This step translates our observations and analogies into concrete operational hypotheses that can guide system design.

Thought Step 3.1: Develop Operational Hypotheses

Based on the emergent dynamics we observed, we hypothesise:

- Hypothesis 1: The train's pantograph interaction with the power line is not passive but introduces dynamic oscillations that propagate along the power line.
- Hypothesis 2: Sparking events, although brief, could interfere with nearby autonomous perception systems, leading to the misclassification of events.
- Hypothesis 3: The power line's oscillatory response to pantograph-induced forces could indirectly affect other infrastructure elements (e.g., sensors, communication networks).

Emergent Scenario Prediction (n26)

We characterise, at least, the following new prediction:

n26: Train sparks and wobbles electric powerline wires [Lateral perspective with respect to train pantograph connection].

This prediction accounts for the non-trivial consequences of pantograph-induced powerline motion, which were previously underappreciated in conventional system models.

Final Summary: How the Thought Process Led to n26

By systematically applying the three-step thought process, we uncovered hidden interactions between the train's pantograph and the electric powerline, leading to the new prediction (n26).

1. Broadening Perspective:
    - Challenged the assumption that powerlines are purely static energy providers by considering dynamic mechanical and electrical interactions.
2. Generating Analogies:
    - Compared train-powerline interactions to guitar string vibrations and tire-road contact loss, refining our understanding of oscillatory effects.
3. Formulating Hypotheses and Scenarios:
    - Identified that pantograph-induced oscillations propagate along the powerline, potentially interfering with perception systems and affecting other infrastructure elements.
    - Developed prediction (n26), establishing an unexpected but safety-relevant interaction.

This systems-thinking approach helps make sense of harder-to-imagine interactions are accounting for them.

H.4.2.2 Step 2.2) Predict the contributing factors (unsafe situations or opportunities)

In this step, we re-write the set of interactions we discovered that constitute complicatedness and categorised them into two contributing factors categories:

- Unsafe problematic activities
- Beneficial / Non-problematic activities.

We do so by examining every interaction and elaborating on its aspects and anticipated impact. Unsafe problematic activities are clear problems that, whatever design solution is chosen, will require further analysis.

Table H.8 Classification of unsafe and beneficial situations in train and adversarial drone interactions (sample of interactions)

Unsafe problematic activities	Beneficial or non-problematic activities
1. Train: 1.1. Unsafe approach of Train to an adversarial drone. Which may lead to striking the train [derived from n6]. 1.4. The train induces random vegetation motion around the fence, which may impede any trained computer vision agent's perception capability [derived from n21]. 1.5. Connected train pantograph to electric powerlines (sparks and wobble). Which may impede the ability to perceive any trained computer vision agents [derived from n26]. **2. Adversarial Drone:** 2.1. Unsafe intended adversarial drone approach train, which may lead to striking the train [derived from n1]. 2.2. Adversarial drones follow train tracks, which may lead to striking the train [derived from n12]. 2.3. Adversarial drone recognises then crosses over the train track zone fence, which may lead to striking the train [derived from n17]. 2.4. Adversarial drone avoids collision with vegetation around the fence, which may lead to striking the train [derived from n22]. 2.6. Adversarial drone avoids electric powerlines, which may lead to a train crash [derived from n22]. 2.7. Adversarial drone collides with electric powerlines, which may lead to train delays [derived from n22].	**1. Train:** 1.2. Maintained train travel over train tracks [derived from n11]. 1.3. The train follows a path within the train tracks' zone fence boundary [derived from n16]. **2. Adversarial Drone:** 2.5. Adversarial drone collides with vegetation around the fence [lateral perspective, for the point of view of inherent defence]

H.4.2.3 Step 2.3) Design problem selection

This step will review the original problem and re-evaluate it. By re-evaluation, we mean we want to be more definite in what set of sub-problems we want to further tackle and which problems we choose to leave. In this step, we will classify the output from step 2.2 into two categories and justify our choice:

- Problems within the design sphere of concern. We will use the following label “To be solved”.
- Situations outside the design sphere of concern. We will use the following label “To be dropped”.

The justification of the design problem sets out the initial high-level problems' context and motivation. The design team may contest some design problem choices, an important activity. Also, it opens up a door to regulators or design assurance teams to seek justifiable reasons why some identified issues are being dropped. At this stage, the design team can apply ethical considerations, which then go towards the trustworthiness case. The following is an example of defining problems:

Table H.9 Design Problem Selection Example

Interaction	Potential Concern	Decision	Elaboration / Justification
n6 (Train 1.1)	The train “unsafely” approaches an adversarial drone, which may lead to striking the train.	To be solved	Even though the train’s path is fixed, collisions can occur if the drone enters the train’s trajectory. The design must address detection or deterrence of drones.
n21 (Train 1.4)	A moving train induces random vegetation motion around the fence, potentially disrupting computer vision perception.	To be solved	This is a direct reliability concern for visual-based detection/monitoring systems. Mitigating vegetation movement effects is critical for robust perception.
n26 (Train 1.5)	Connected train pantograph with electric powerlines (sparks, wobble) may impede perception or cause transient noise.	To be dropped	While sparks do happen, they are infrequent and minor. The design team judges it as low priority compared to other, more frequent hazards.

n1 (Drone 2.1)	Adversarial drone intentionally approaches the train, risking collision.	To be solved	A primary safety hazard. The design must detect/prevent drones from dangerously encroaching upon the train's path.
n12 (Drone 2.2)	Adversarial drones follow train tracks, risking eventual collision with the train.	To be solved	Tracking the train's route is a plausible tactic for an adversarial drone. System solutions must anticipate route-based drone threats.
n17 (Drone 2.3)	Adversarial drone recognizes, then crosses over the train track fence, potentially striking the train.	To be solved	One of the core unsafe situations. The design will focus on detecting/preventing unauthorized intrusion into the fenced train track zone.
n22 (A) (Drone 2.4)	Drone tries to avoid collision with vegetation, but inadvertently risks striking the train.	To be solved	The drone's avoidance maneuvers create a new collision vector with the train. The design must handle or predict these emergent drone movements.
n22 (B) (Drone 2.6)	Drone avoids electric powerlines, creating a new path that could cause a train crash.	To be solved	Avoiding powerlines is good for the drone, but poses serious risk if it diverts into the train's path. Must be addressed in hazard avoidance strategies.
n22 (C) (Drone 2.7)	Drone mildly collides with electric powerlines, leading to train delays (instead of a crash).	To be dropped	Though undesirable, it is a lesser impact scenario (delay vs. collision). The design scope primarily focuses on preventing collisions that endanger train safety.
(Beneficial) n11 (Train 1.2)	Normal train travel over tracks (i.e., operating as intended).	To be dropped	This is standard and safe operation, not a "problem" requiring a design fix.
(Beneficial) n16 (Train 1.3)	The train follows its path within the fenced boundary.	To be dropped	Again, this is expected safe behavior with no direct hazard to mitigate.

(Beneficial) Drone 2.5	Adversarial drone collides with vegetation around the fence (could be inherently protective for the train).	To be dropped	This scenario may even be beneficial for train safety (the drone is stopped by vegetation). No direct hazard or design fix is required.

H.4.3 Predictive Thinking Pipeline 3: Predict the Emergence of AIC Complexity Field for Detailed Operational Scenario Articulation

Predictive Thinking Pipeline 2 helped us define the potential problems that we need to solve and the choices of problems to solve. In this pipeline, we will dive deeper into the AIC complexity of each interaction discovered and then identify the intricate relationships.

H.4.3.1 Step 3.1) Model detailed AIC interactions scenarios for the problem domain

For every unsafe, problematic, or beneficial interaction, and considering the source complex's PrimeP, visualise the emergence of AIC Complicated Behaviour using an AIC modelling schema. In this case, any factor modelled must be written with a specific situation in the format of {adjective_name}. For example, if you define train as a factor, a situation needs to be mentioned with it, for instance {moving_train} or {roaming_adversarial_drone}. to do so, we need to define a conventional way to refer to any complex of interest.

Being clear about a factor's dynamic or static situation is a critical thinking step to resolve the complicatedness of any complexity. AIC modelling schema models the dynamic and static situations of systems. Also, actions are to be modelled as verbal phrases and signed (-) if they carry an intended obstructive goal, (+) if they carry an intended supportive goal, or unsigned if they carry a neutral unintended impact.

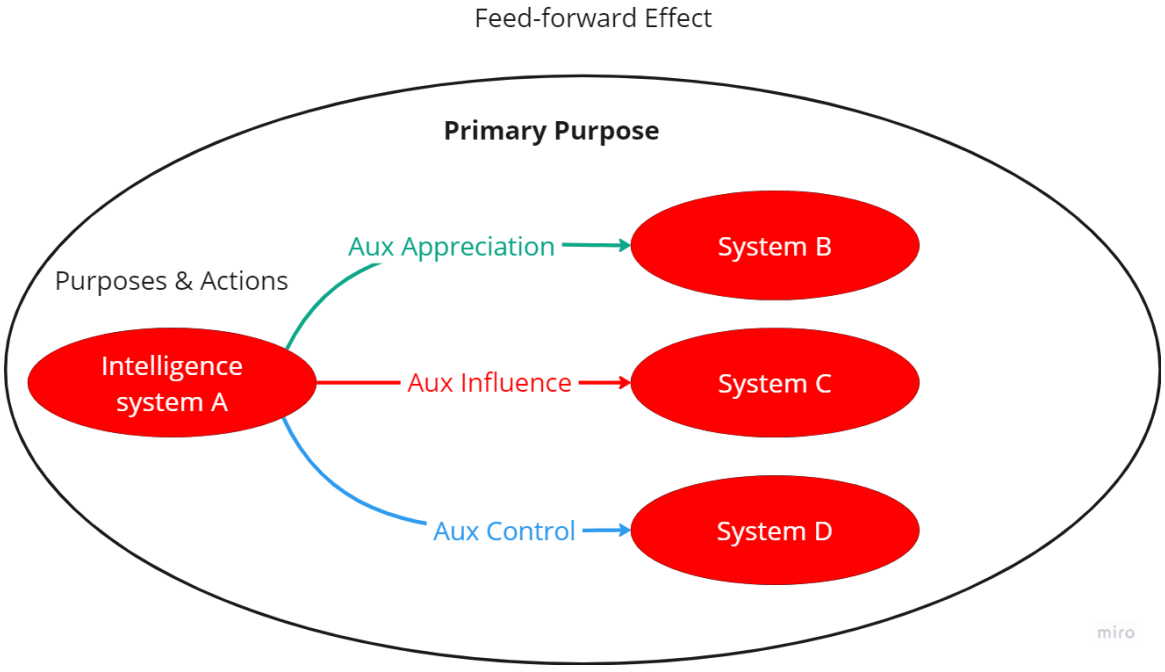


Figure H.1 Forward-Feed partial AIC-SECoT

For example, consider adversarial drone PrimeP: disrupt trains operations. Our case study will consider only the Forward-Feed partial AIC CoT. We will choose the unsafe approach of a Train to an adversarial drone, which may lead to striking the train [derived from n6].

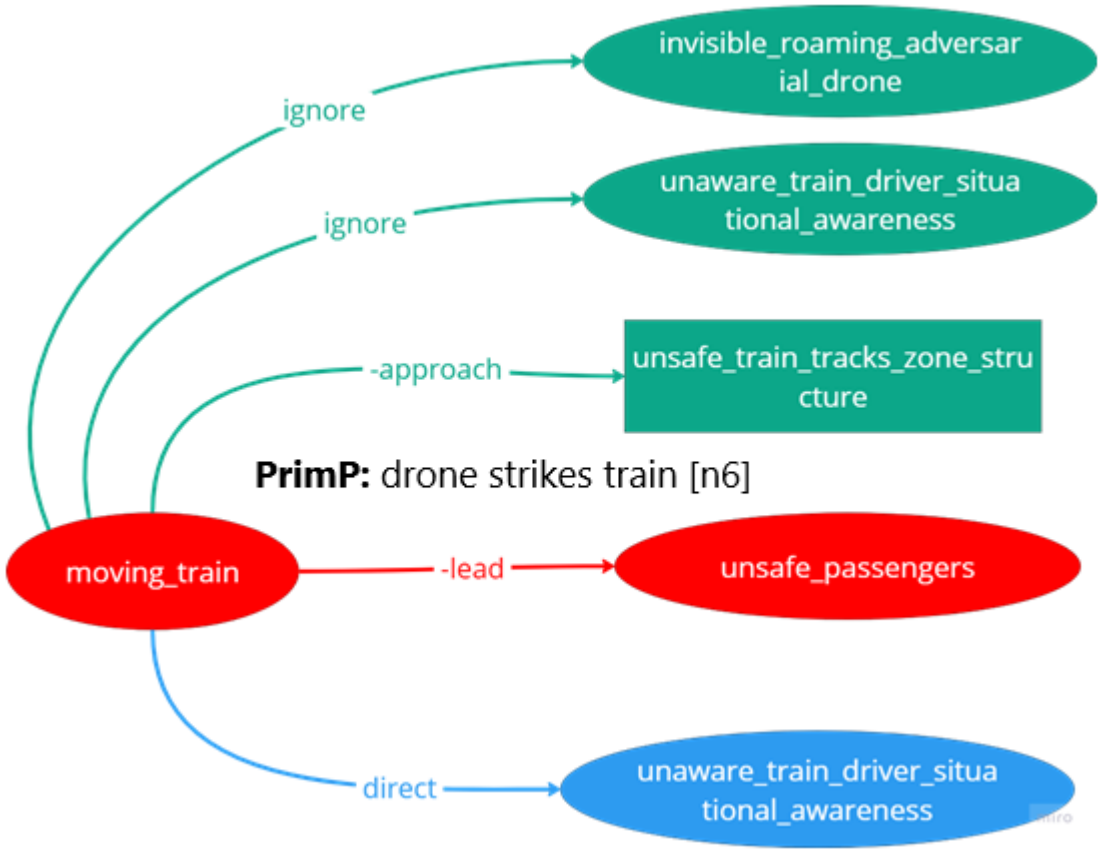


Figure H.2 Modelling a complicated interaction n6

H.4.3.2 Step 3.2) Predict the extended list of emergent AIC interactions scenarios.

We then capture the interactions in the figure above using the following SECoT:

Table H.10 Complexity Field for n6 Interaction SECoT definition

Step 1: Unsafe problematic activities	Unsafe approach of Train to an adversarial drone. Which may lead to striking the train [derived from n6].	
Step 2: Observed System (obs)	**Step 3: Observed Action**	**Step 4: supra source Primary Purpose**
roaming_adversarial_drone, train_derailment	Roaming adversarial drone approach train	Adversarial Scheme PrimeP: Disrupt Train Network operations.
Step 5: Auxiliary Influence Interaction	**Step 6: Auxiliary Control Interaction**	**Step 7: Auxiliary Appreciation Interaction**
|{moving_train}_[-lead]_{unsafe_unsecure	|{moving_train}_[+direct]_{unaware_train_driver_situational_awareness}|	|{moving_train}_[approach]_{unsafe_train_tracks_zone_structure}|

d_passengers_and_goo ds}|		|{moving_train}_[ignore]_{unawar e_train_driver_situational_aware ness}| |{moving_train}_[ignore]_{roaming _adversarial_drone}|
Step 8: Predicted Problem Domain Factors or Features (with repetition)		
Appreciation = [Train_Network, moving_train, unsafe_train_tracks_zone_structure, train_transit_through_track_zone] Influence = [moving_train, unsafe_unsecured_passengers_and_goods] Control = [moving_train, unaware_train_driver_situational_awareness]		

Table H.10 outlines the complicated interaction between a train and an adversarial drone within a constrained train track zone. This SECoT process begins by identifying **Unsafe problematic activities** (Step 1), specifically the unsafe approach of a train to a roaming adversarial drone, which could lead to derailment and significant safety and operational risks. The problematic situation is derived from the n6 interaction, emphasising the interplay between the drone's disruptive potential and the train's vulnerable transit through an exposed zone.

The **observed system (obs)** (Step 2) includes critical actors and systems such as the roaming adversarial drone and the potential for train derailment. In Step 3, **observed actions** are mapped, detailing the adversarial drone's approach towards the train, illustrating its intentional disruption of operations. The **supra source primary purpose** (Step 4), derived as the *Adversarial Scheme Prime Purpose (PrimeP)*, is to disrupt train network operations, framing the drone as a purposeful actor within the broader complexity field.

Subsequent steps decompose interactions into three dimensions: **auxiliary influence, control, and appreciation interactions** (Steps 5–7). Influence interactions capture how the moving train affects unsafe conditions, such as unsecured passengers and goods. Control interactions map the train's lack of situational awareness due to an uninformed driver, exacerbating vulnerability. Appreciation interactions consider the train's approach to the unsafe track zone structure, which is further complicated by the invisible presence of the drone. The interplay between ignoring critical threats and neglecting situational awareness is a key failure in system-level feedback.

Lastly, **predicted problem domain factors** (Step 8) consolidate the identified elements into three dimensions of systemic engagement. Appreciation emphasises structural and operational aspects such as the train network, moving train, and unsafe track zone structure. Influence focuses on the broader impact on unsecured passengers and goods, while control centres on the

train's operational deficiencies, particularly the driver's situational awareness. These insights provide a robust framework for understanding compounded risks and designing targeted interventions within the complexity field of train-track adversarial dynamics.

Let's take another example, we will model n12 interaction:

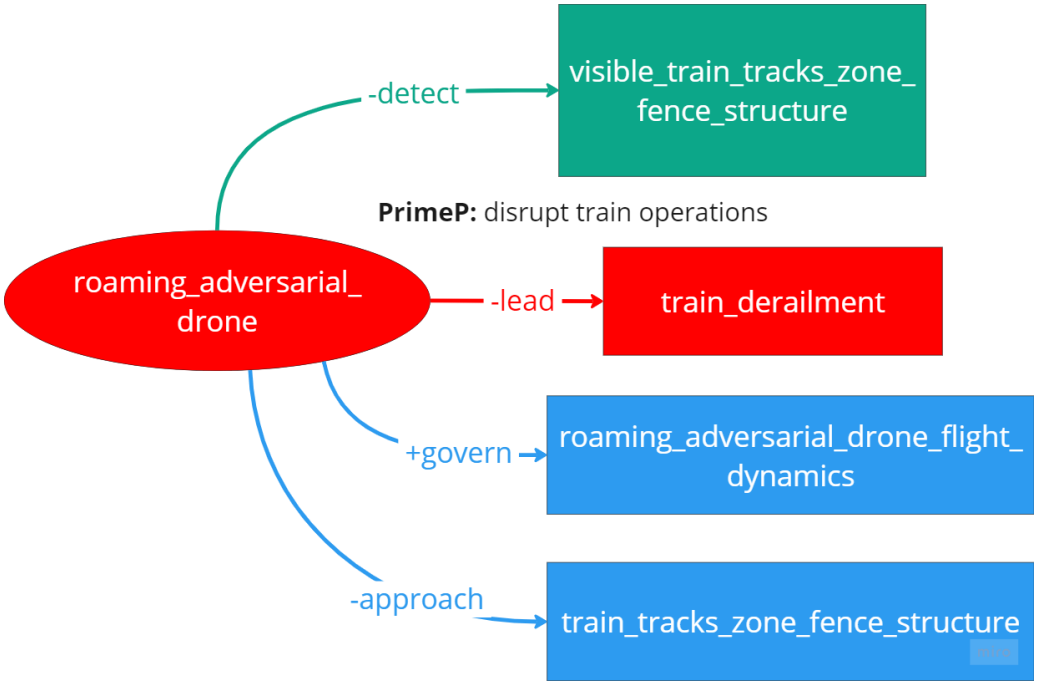


Figure H.3 AIC complexity field for n12 interaction

The following is the SECoT definition of the model:

Table H.11 Adversarial drone follows the train tracks n12 SE-CoT.

Step 1: Unsafe problematic activities	Adversarial drones follow train tracks, which may lead to striking the train [derived from n12].	
Step 2: Observed System (obs)	**Step 3: Observed Action**	**Step 4: supra source Primary Purpose**
roaming_adversarial_drone, train_derailment	Roaming adversarial drone approach train	Adversarial Scheme PrimeP: Disrupt Train Network operations.
Step 5: Auxiliary Influence interaction	**Step 6: Auxiliary Control interaction**	**Step 7: Auxiliary Appreciation interaction**
|{roaming_adversarial_drone}_[-lead]_{train_derailment}|	|{roaming_adversarial_drone}_[+govern]_{roaming_adversarial_drone_flight_dynamics}|,|{roaming_adversarial_drone}_[-approach]_{train_tracks_zone_fence_structure}|	|{roaming_adversarial_drone}_[-detect]_{visitble_train_tracks_zone_visible_structure}|
Step 8: Predicted Problem Domain Factors or Features (with repetition)		

Appreciation = ['roaming_adversarial_drone', 'visitble_train_tracks_zone_visible_structure']
Influence = ['roaming_adversarial_drone', 'train_derailment']
Control =['roaming_adversarial_drone', 'roaming_adversarial_drone_flight_dynamics', 'roaming_adversarial_drone', 'train_tracks_zone_fence_structure']

Table H.11 examines the unsafe scenario in which adversarial drones follow train tracks, potentially leading to a catastrophic train derailment. The analysis begins by identifying the **unsafe problematic situation** (Step 1), focusing on the adversarial drone's capacity to follow and disrupt the train's operations. This scenario highlights a deliberate intrusion aimed at exploiting vulnerabilities in the train's operational environment, particularly the track zone, to achieve the malicious goal of derailment.

The **observed system (obs)** (Step 2) includes the roaming adversarial drone and the potential for train derailment, which is exacerbated by the drone's behaviour, as detailed in Step 3. The **observed action** involves the adversarial drone's strategic approach toward the train, further emphasising its intent to disrupt operations. This is aligned with the **supra source primary purpose** (Step 4), defined as the *Adversarial Scheme Prime Purpose (PrimeP): Disrupt Train Network operations*. This highlights the drone's role as a purposeful agent in undermining the safety and functionality of the train network.

Steps 5 to 7 decompose the situation into **auxiliary interactions** that characterise the drone's influence, control, and appreciation of its operational environment. **Influence interactions** describe the negative impact of the drone's actions, such as leading to derailment. **Control interactions** address the drone's governed flight dynamics and its interaction with physical structures, such as the train track zone fence. The **appreciation interaction** focuses on the drone's ability to perceive and adapt to visible elements of the train track zone structure, underscoring its capability to navigate and exploit the environment.

Step 8 synthesises the **predicted problem domain factors** into three dimensions: appreciation, influence, and control. **Appreciation factors** include the roaming adversarial drone and the visible structures of the train track zone, highlighting the environmental elements critical for situational awareness. **Influence factors** focus on the drone's ability to affect the train's safety, particularly the risk of derailment. **Control factors** address the drone's governance over its flight dynamics and its interaction with physical barriers like fences. These insights underscore the complexity of adversarial drone threats and provide a structured basis for developing mitigation strategies that address the multifaceted risks within this operational context.

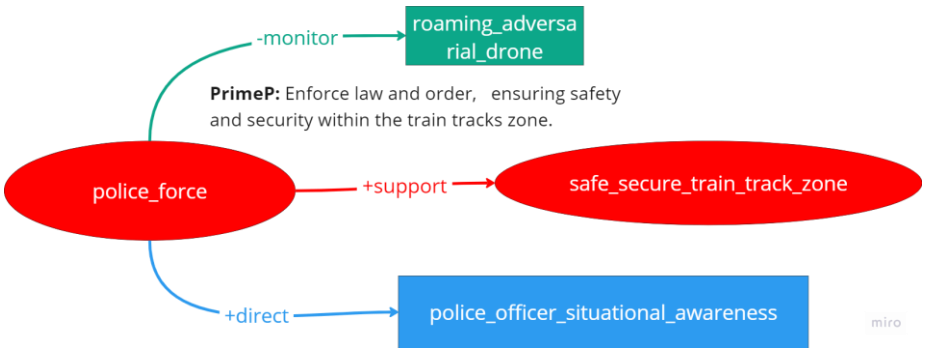


Figure H.4 Operational scenario derived from n 32

The following is the SECoT definition of the model:

Table H.12 Police Force Response to Adversarial Drones in Train Track Zones

Step 1: Unsafe problematic activities	Police Officers are incapable of catching adversarial drones.[derived from n32].	
Step 2: Observed System (obs)	**Step 3: Observed Action**	**Step 4: supra source Primary Purpose**
roaming_adversarial_drone, police officer	police officer incapable to capture Roaming adversarial drone	Police Force PrimeP: Enforce law and order, ensuring safety and security within the train tracks zone.
Step 5: Auxiliary Influence interaction	**Step 6: Auxiliary Control interaction**	**Step 7: Auxiliary Appreciation interaction**
|{police_officer_situational_awareness}_[+support]_{safe_secure_train_track_zone}|	|{police_force}_[+direct]_{police_officer_situational_awareness}|	|{police_force}_[-monitor]_{roaming_adversarial_drone}|
Step 8: Predicted Problem Domain Factors or Features (with repetition)		
Appreciation = ['roaming_adversarial_drone'] Influence = ['safe_secure_train_track_zone'] Control =['police_officer_situational_awareness']		

Table H.12 analysis highlights the unsafe situation where police officers are incapable of capturing roaming adversarial drones, compromising safety within the train track zone. The observed system and action focus on the police force's inability to monitor and respond effectively to drone threats. The supra source's Primary Purpose (PrimeP) is to enforce law and order, ensuring a secure train track environment. Auxiliary interactions emphasise the police force's need to enhance situational awareness (control), support the train track zone's safety

(influence), and effectively monitor adversarial drones (appreciation). This analysis identifies critical factors such as the roaming adversarial drone, the need for situational awareness, and maintaining a secure zone to address this challenge comprehensively.

H.4.3.3 Step 3.3) Collate factors/situations (step 8 in the table)

Collate all factors in between the curly brackets {} and capture them in AIC lists. Include the following information types: Source, Sink, and Supra Systems. For example;

Appreciation = [Train_Network, moving_train, unsafe_train_tracks_zone_structure, train_transit_through_track_zone]

Influence = [moving_train, unsafe_unsecured_passengers_and_goods]

Control = [moving_train, unaware_train_driver_situational_awareness]

H.4.4 Predictive Thinking Pipeline 4: Predict and Evaluate Problem Domain Factors and Assumptions.

H.4.4.1 Step 4.1) Perform most and least frequent factor evaluation

Evaluate AIC factors and list them without repetition from the most frequent factor to the least frequent. We extracted the factors by running a Python script and listed the text in between {x}. Then we counted their frequency of mention, which indicated our problem perception bias. The most common factors are those we view as the most relevant, while the least common factors are those we don't usually consider. For example,

1. roaming_adversarial_drone
2. moving_train
3. train_tracks_structure

Factors frequency evaluation

The fascinating finding in this analysis is that the frequency distribution of the factors followed a power-law distribution, a trend commonly observed in black swan events and heavy-tailed distributions. This trend line created a curve that is easily fitted, a power law distribution curve with 86% fitness. This also indicates that the analysis produced a reasonable result, as the output predicted a reasonable set of factors corresponding to an anticipated probability distribution.

Table H.13A Predicted Factors Output = 94 factors

Predicted Factor	Number of mentions	Concern Level
roaming_adversarial_drone	65	20.00%
moving_train	58	17.85%
train_tracks_structure	10	3.08%
any_trained_computer_vision_agent_perception_capability	8	2.46%
train_tracks_zone_fence_structure	8	2.46%
train_derailment	7	2.15%
fallen_vegetation_over_train_tracks_structure	7	2.15%
powered_powerlines_cables_structure_visual_appearance	6	1.85%
vegetation_complexity_growth	6	1.85%
police_officer_situational_awareness	6	1.85%
train_driver_situational_awareness	5	1.54%
available_ground_based_security_systems	5	1.54%
roaming_adversarial_drone_perception_capability	5	1.54%
roaming_adversarial_drone_flight_dynamics	5	1.54%
onboard_train_safety_systems	4	1.23%
extended_train_pantograph	4	1.23%
train_tracks_structure_visual_appearance	4	1.23%
police_officer_capture_adversarial_drone	4	1.23%
unsafe_train_tracks_zone_structure	3	0.92%
unsafe_unsecured_passengers_and_goods	3	0.92%
alert_systems	3	0.92%
roaming_adversarial_drones_location_data	3	0.92%
weather_conditions	3	0.92%
free_space_above_train_roof	3	0.92%
vegetation_complex_shapes_sizes_visual_appearance	3	0.92%
visible_powered_powerlines_cables_structure_visual_appearance	3	0.92%
police_force	3	0.92%
stationary_train	3	0.92%
unaware_train_driver_situational_awareness	2	0.62%
safe_passengers_and_goods	2	0.62%
roaming_adversarial_drones_information	2	0.62%
unaware_emergency_response_teams	2	0.62%

onboard_train_communication_interface	2	0.62%
invisible_roaming_adversarial_drone_visual_appearance	2	0.62%
camouflage_mechanisms	2	0.62%
train_schedule_timing	2	0.62%
train_tracks_maintenance_activities	2	0.62%
track_zone_obstacles	2	0.62%
visible_train_tracks_zone_fence_structure_visual_appearance	2	0.62%
passing_train_season_&_wind	2	0.62%
safe_secure_train_track_zone	2	0.62%
track_zone_monitoring_systems	1	0.31%
track_zone_monitoring_systems_data	1	0.31%
timely_track_zone_monitoring_systems_data	1	0.31%
accurate_track_zone_monitoring_systems_data	1	0.31%
track_zone_monitoring_systems_data_gaps_or_inconsistencies	1	0.31%
roaming_adversarial_drones	1	0.31%
roaming_adversarial_drones_size_data	1	0.31%
roaming_adversarial_drones_flying_behaviour_data	1	0.31%
roaming_adversarial_drones_critical_data	1	0.31%
roaming_adversarial_drones_information_transmitted_data	1	0.31%
designated_emergency_response_communication_system	1	0.31%
emergency_notification	1	0.31%
reliable_railways_communication_network	1	0.31%
accurate_track_zone_monitoring_systems	1	0.31%
communication_network_connectivity	1	0.31%
alternative_communication_channels	1	0.31%
track_zone_monitoring_systems_data_plausibility	1	0.31%
ground_based_security_systems_received_data	1	0.31%
disruptive_adversarial_drones_threat_severity	1	0.31%
low_altitude_roaming_adversarial_drone	1	0.31%
roaming_adversarial_drone_camouflage	1	0.31%
roaming_adversarial_drone_noise_emissions	1	0.31%
moving_train_blind_spots	1	0.31%
altitude_control_systems	1	0.31%
roaming_adversarial_drone_noise_suppression_systems	1	0.31%
roaming_adversarial_drone_navigation_systems	1	0.31%

altitude_parameters	1	0.31%
roaming_adversarial_drone_noise_output	1	0.31%
path_planning_algorithms	1	0.31%
adversarial_drone_response_strategies	1	0.31%
flight_path_based_on_weather	1	0.31%
routes_around_maintenance_activities	1	0.31%
adversarial_drone_dangerous_navigation_around_obstacles	1	0.31%
random_vegetation_motion	1	0.31%
traveling_train_induced_air_turbulence_and_vibration	1	0.31%
fence_to_train_open_space	1	0.31%
train_electric_power_supply	1	0.31%
powered_powerlines_cables_structure_sparks	1	0.31%
visible_train_tracks_appearance	1	0.31%
visible_train_tracks_zone_fence_structure	1	0.31%
visible_vegetation_visual_appearance	1	0.31%
visible_vegetation_structure	1	0.31%
powered_powerlines_cables_EMF	1	0.31%
powered_powerlines_cables_structure	1	0.31%
visibility_obstructive_objects	1	0.31%
adversarial_drone_collision_with_train	1	0.31%
train_track_zone_structure	1	0.31%
leaves_&_pollen	1	0.31%
passing_train_approach	1	0.31%
safe_secure_police_officer	1	0.31%
all_trains_operations	1	0.31%
police_officer_movements	1	0.31%

The table shows the top 5 factors with the frequency of being mentioned equal or more than 8 times. These factors are rather obvious for the architect. However, there are 89 factors that have been mentioned 7 times or less. All of which were the hidden black swan events of the problem domain. In this step, evaluate each factor's repetition frequency and calculate the influence level. The term "roaming_adversarial_drone" has been cited 65 times out of 328 factors (including repetitions). This means the influence level of "moving_train" is roughly 18% of the total influence, calculated as 65 divided by 328. Similar to moving_train, which has a 17% influence on the architect's perception. This indicates that our overall architect’s perception of the problem pays much more bias towards roaming_adversarial_drone, and the bias had blinded their perspective

to notice other hidden factors which the analysis had helped to reveal. Thus appearing to be the most important factor in the problem domain is not whether an adversarial drone is accessing the train track zone but whether the train is in motion or not.

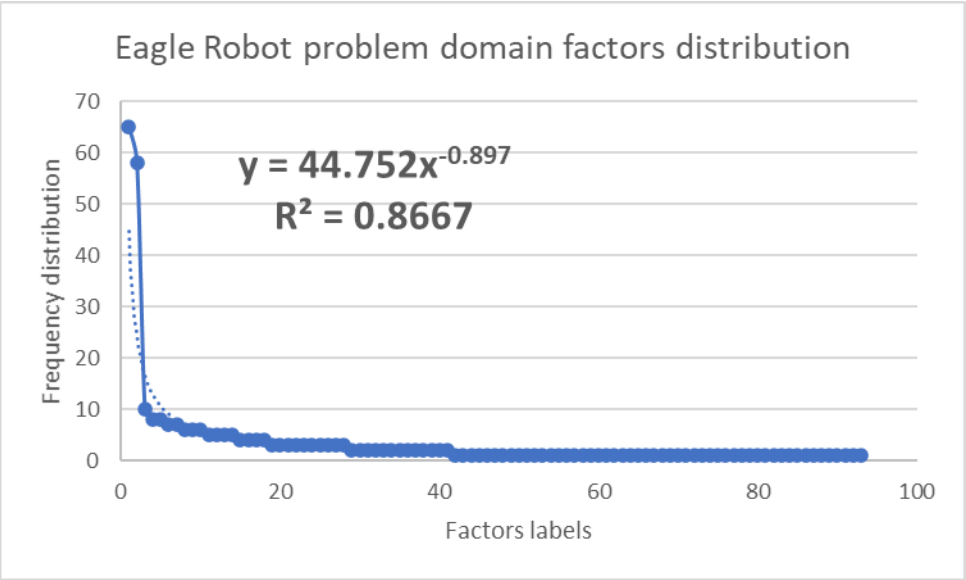


Figure H.5 Frequency distribution curve of the derived factors

The following is an example of factors that only occurred once, which indicates factors that we are not paying too much attention to:

- **leaves_&_pollen:** falling leaves and pollen carried in the air, which impacts the reliability of autonomous systems.
- **open_fence_to_train_space:** the space between the fence and the moving train.
- **routes_around_maintenance_activities:** if maintenance engineers are present, the route between their activities can be open for adversarial drones to get access to.

These factors were not part of the initial problem brief, but the process helped us identify them objectively. We define those factors as harder-to-predict relative to our perception of the problem at the beginning of the analysis.

H.4.4.2 Step 4.2) Define all identified problem domain factors.

Define each factor. We identify 93 types of factors involved within the complicatedness of the problem complexity. For example.

roaming_adversarial_drones: Refers to roaming adversarial drones and the information related to them, particularly their location, information, size, flying behaviour, and critical data are communicated, filtered, or prioritised during operational scenarios.

powered_powerlines_cables_structure_visual_appearance: Refers to the visual appearance and condition of powered powerlines and cable structures, particularly in how they are affected or complicated by the presence and interactions with train tracks and related structures.

Table H.14 Problem domain factors definitions

Predicted Factor	**Factor Definition**
moving_train	Refers to a train that is in motion along its designated tracks, involved in various actions such as approaching unsafe zones, maintaining safety, and communicating with onboard systems.
roaming_adversarial_drone	Represents a hostile drone roaming around, possibly gathering information, affecting communication, and potentially causing disruptions or dangers to train operations.
train_tracks_structure	Indicates the physical structure and layout of train tracks, influencing visibility, obstacles, and potential collisions or obstructions encountered during train operations.
any_trained_computer_vision _agent_perception_capability	Refers to the capability of computer vision agents to perceive obstacles or conditions along train tracks, impacted by factors like vegetation growth or train structures.
train_tracks_zone_fence_structure	Describes the fencing and demarcation around train tracks, influencing the perception and actions of adversarial drones and the safety of train operations.
fallen_vegetation_over_train_ tracks_structure	Refers to vegetation that has fallen onto train tracks, potentially impacting the movement or visual perception of the train tracks.
train_driver_situational_awareness	Refers to the level of awareness and understanding that a train driver has about their operational environment, influencing their decision-making and responsiveness.
powered_powerlines_cables_ structure_visual_appearance	Refers to the visual appearance or condition of powerlines and cables that are installed near or along train tracks, affecting their visibility or operational safety.

train_derailment	Refers to the instance where a train leaves its tracks, often due to external factors or conditions that disrupt normal operation.
vegetation_complexity_growth	Refers to the growth and complexity of vegetation near train tracks, influencing various aspects such as visibility, operational complexity, and environmental impact.
police_officer_situational_awareness	Refers to the awareness and readiness of police officers regarding their operational environment, specifically in relation to ensuring safety and security around train tracks and associated areas.
unsafe_unsecured_passengers_and_goods	Refers to the presence of passengers or goods that are not properly secured or pose a safety risk when transported on a moving train, potentially leading to safety hazards or operational disruptions.
onboard_train_safety_systems	Refers to the integrated systems and mechanisms aboard a train designed to enhance safety, control, and operational efficiency during transit and in response to various environmental and operational conditions.
available_ground_based_security_systems	Refers to the range of security systems and protocols that operate on the ground to monitor, communicate, and ensure the safety and security of trains, passengers, and infrastructure in transit environments.
roaming_adversarial_drone_ perception_capability	Refers to the capability of adversarial drones to perceive and interact with their environment, specifically in relation to their ability to detect, navigate, and potentially disrupt train operations and infrastructure.
roaming_adversarial_drone_flight_dynamics	Refers to the dynamics and behavior of adversarial drones during flight, particularly

	in how they are governed and controlled to influence their operational impact.
unsafe_train_tracks_zone_structure	Refers to structural elements near train tracks that are deemed unsafe, potentially posing risks or hazards to trains approaching or passing through the area.
alert_systems	Refers to systems designed to alert or notify relevant parties about specific conditions or events related to train operations, safety, or security.
extended_train_pantograph	Refers to the extended apparatus on trains that connects to overhead power lines, impacting both operational capabilities and potential risks related to infrastructure and safety.
train_tracks_structure_visual_appearance	Refers to the visual appearance and condition of train tracks and associated structures, influencing operational safety, maintenance needs, and environmental interactions.
police_officer_capture_adversarial_drone	Refers to the actions and strategies employed by police officers to capture or mitigate the impact of adversarial drones in operational environments.
roaming_adversarial_drones_location_data	Refers to the data related to the location and movements of adversarial drones, which is crucial for monitoring and responding to potential threats or disruptions.
weather_conditions	Refers to the current and anticipated atmospheric conditions that may impact train operations, safety, and the behavior of adversarial drones.
free_space_above_train_roof	Refers to the airspace clearance above the roof of a moving train, influencing the ability of drones or other objects to safely navigate or pose threats.

vegetation_complex_shapes_ sizes_visual_appearance	Refers to the visual characteristics and appearances of vegetation, including shapes and sizes, particularly in how they complicate or affect the visual environment around train tracks.
visible_powered_powerlines_cables _structure_visual_appearance	Refers to the visual appearance and condition of powered powerlines and cables structures, particularly in how they are affected or complicated by the presence and interactions with train tracks and related structures.
police_force	Refers to the collective body of law enforcement personnel and their organized efforts in response to or in support of various operational scenarios involving adversarial drones and other safety or security concerns.
stationary_train	Refers to a train that is not in motion, impacting its interactions and potential impediments posed to police officer movements or actions regarding adversarial drones and other operational challenges.
unaware_train_driver_situational_awareness	Refers to the level of awareness or lack thereof among train drivers regarding their operational surroundings and potential hazards, particularly influenced by factors like vegetation complexity and its impacts.
safe_passengers_and_goods	Refers to ensuring the safety and security of passengers and cargo aboard trains, maintaining their well-being and minimizing risks associated with operational impacts or adverse scenarios.
track_zone_monitoring_systems	Refers to systems designed for monitoring and data collection related to specific track zones, ensuring timely and accurate information for operational decision-making

	and response to potential hazards or disruptions.
roaming_adversarial_drones_information	Refers to the information gathered and transmitted by adversarial drones, particularly in how it is communicated or utilized in operational environments such as train tracks and associated safety or security measures.
unaware_emergency_response_teams	Refers to emergency response teams unaware of specific operational impacts or challenges, potentially triggered into action by critical events or communications related to train operations or adversarial drone incidents.
onboard_train_communication_interface	Refers to the interface or system aboard trains used for transmitting data packets or emergency notifications, ensuring effective communication and response capabilities during operational incidents or emergencies.
low_altitude_roaming_adversarial_drone	Refers to adversarial drones operating at low altitudes, particularly in how they maintain their presence and potential impacts in scenarios involving invisible roaming adversarial drones.
roaming_adversarial_drone_camouflage	Refers to the use of camouflage by invisible roaming adversarial drones, particularly in how they employ such tactics to blend into their environment or avoid detection.
roaming_adversarial_drone_noise_emissions	Refers to the noise emissions from invisible roaming adversarial drones, particularly in how they are suppressed or managed to minimize their impact or detection during operational scenarios.
moving_train_blind_spots	Refers to blind spots around moving trains that adversarial drones exploit, influencing operational safety and security measures

	related to their movements and potential threats they pose.
camouflage_mechanisms	Refers to mechanisms that control or activate camouflage for invisible roaming adversarial drones, particularly in how they are employed to enhance stealth capabilities and operational effectiveness in various environments.
train_schedule_timing	Refers to the timing synchronization and appreciation related to train schedules, particularly in how invisible roaming adversarial drones may influence or synchronize with these schedules.
train_tracks_ maintenance_activities	Refers to activities related to the maintenance of train tracks, particularly in how they are appreciated or scanned for potential impacts from invisible roaming adversarial drones.
track_zone_obstacles	Refers to obstacles within track zones, particularly in how they are appreciated or detected for potential interference or risks posed by invisible roaming adversarial drones.
visible_train_tracks_zone_fence_ structure_visual_appearance	Refers to the visual appearance of structures like fences around visible train tracks, particularly in how they are complicated or ignored in relation to the operational impacts described involving train tracks.
passing_train_season_&_wind	Refers to seasonal and wind factors influencing the passage of trains, particularly in how they follow or are influenced by fallen vegetation over train tracks, affecting operational scenarios involving train schedules and safety measures.
safe_secure_train_track_zone	Refers to ensuring safety and security within train track zones, particularly supported by

	police force actions and situational awareness measures in response to operational challenges involving adversarial drones.
track_zone_monitoring_systems_data	Refers to data collection initiated by moving trains for track zone monitoring systems, particularly in how it is acknowledged, highlighted for gaps or inconsistencies, or verified for plausibility in operational environments.
timely_track_zone_ monitoring_systems_data	Refers to timely data acknowledgment within track zone monitoring systems, particularly in how it is appreciated or synchronized with operational needs involving moving trains.
accurate_track_zone_ monitoring_systems_data	Refers to accurate data acknowledgment within track zone monitoring systems, particularly in how it is acknowledged or appreciated for its reliability and relevance to moving train operations.
track_zone_monitoring_ systems_data_ gaps_or_inconsistencies	Refers to gaps or inconsistencies in data within track zone monitoring systems, particularly in how they are highlighted or acknowledged for potential operational impacts and improvements.
roaming_adversarial_drones	This refers to roaming adversarial drones and the information related to them, particularly their location, information, size, flying behaviour, and critical data are communicated, filtered, or prioritized during operational scenarios.
roaming_adversarial_drones_size_data	Refers to data related to the size of roaming adversarial drones, particularly in how it is filtered or managed for operational purposes involving their movements and potential threats.

roaming_adversarial_drones_ flying_behaviour_data	Refers to data related to the flying behavior of roaming adversarial drones, particularly in how it is filtered or managed for operational purposes involving their movements and potential threats.
roaming_adversarial_drones_critical_data	Refers to critical data related to roaming adversarial drones, particularly in how it is prioritized or managed for operational purposes involving their movements and potential impact assessment.
roaming_adversarial_drones_ information_transmitted_data	Refers to transmitted data related to roaming adversarial drones, particularly in how it is formatted or managed for operational purposes involving their information and impact on operational decisions.
designated_emergency_response_ communication_system	Refers to systems established for emergency response communication, particularly in how they establish connections and ensure effective communication during moving train operations and related incidents.
emergency_notification	Refers to notifications transmitted during emergencies, particularly in how they are transmitted via onboard train communication interfaces to ensure timely responses and actions.
reliable_railways_communication_network	Refers to the appreciation of a reliable communication network within railways, particularly in how it supports operational needs and ensures effective communication during moving train operations.
accurate_track_zone_monitoring_systems	Refers to the acknowledgment of accurate track zone monitoring systems, particularly in how they are appreciated for their reliability and relevance to operational decisions involving moving trains.

communication_network_connectivity	The capability and quality of connections between communication systems relevant to the train.
alternative_communication_channels	Channels established for communication in case of disruptions or failures in primary channels.
track_zone_monitoring_ systems_data_plausibility	The credibility and reliability of data collected by monitoring systems in track zones.
ground_based_security_ systems_ received_data	Data received by ground-based security systems confirming various aspects related to train security.
disruptive_adversarial _ drones_threat_severity	Assessment of the severity level posed by disruptive adversarial drones in a given scenario.
altitude_control_systems	Systems responsible for controlling the altitude of invisible roaming adversarial drones.
roaming_adversarial_drone_noise_ suppression_systems	Systems designed to reduce noise emissions generated by roaming adversarial drones.
roaming_adversarial_drone _navigation_systems	Navigation systems enabling control over the movement and direction of roaming adversarial drones.
altitude_parameters	Parameters adjusted to regulate the altitude of invisible roaming adversarial drones.
roaming_adversarial_ drone_noise_output	Output level of noise generated by roaming adversarial drones in a controlled scenario.
path_planning_algorithms	Algorithms optimized for planning the optimal path of invisible roaming adversarial drones.
adversarial_drone_response_strategies	Strategies adapted to counteract and respond to the actions of invisible roaming adversarial drones.
flight_path_based_on_weather	Adaptation of flight paths of invisible roaming adversarial drones based on current weather conditions.

routes_around_maintenance_activities	Planning of alternative routes for invisible roaming adversarial drones to avoid maintenance activities.
adversarial_drone_dangerous_ navigation_around_obstacles	Navigation protocols for safely maneuvering invisible roaming adversarial drones around obstacles.
random_vegetation_motion	Agitation caused by the movement of vegetation as a neutral impact action of moving trains.
traveling_train_induced_air_ turbulence_and_vibration	Induced turbulence and vibration in the air by traveling trains as a neutral impact action.
fence_to_train_open_space	Interaction or passing of moving trains through open spaces within fenced areas.
train_electric_power_supply	Management and provision of electric power to moving trains as a neutral impact action.
powered_powerlines_ cables_structure_sparks	Incidents where extended train pantographs inadvertently trigger sparks from powered cables and structures.
invisible_roaming_adversarial_ drone_visual_appearance	Management of the visual appearance of roaming adversarial drones to ensure invisibility.
visible_train_tracks_appearance	The visual appearance or clarity of train tracks that is perceptible to roaming adversarial drones.
visible_train_tracks_zone_fence_structure	The structure and appearance of fenced zones around visible train tracks, affecting detection capabilities of roaming adversarial drones.
visible_vegetation_visual_appearance	The visual recognition or appearance of vegetation as observed by roaming adversarial drones.
visible_vegetation_structure	The physical structure or arrangement of vegetation that can be encountered or avoided by roaming adversarial drones.

powered_powerlines_cables_EMF	The electromagnetic fields emitted by powered powerlines and cables, potentially ignored by roaming adversarial drones.
powered_powerlines_ cables_structure	The triggering of sparks and structural aspects related to powered powerlines and cables, potentially recognized or avoided by roaming adversarial drones.
visibility_obstructive_objects	The degree to which obstructive objects are visible to roaming adversarial drones on train tracks.
adversarial_drone_collision_with_train	The occurrence and consequences of collisions between adversarial drones and trains on tracks.
train_track_zone_structure	The structure and configuration of train track zones that can influence adaptation by vegetation complexity and growth.
leaves_&_pollen	The generation or presence of leaves and pollen due to fallen vegetation over train tracks.
passing_train_approach	Monitoring the approach of passing trains as part of supportive actions by police officer situational awareness.
safe_secure_police_officer	Supporting actions by police officers for ensuring safety and security in a given situation.
train_operations	Impeding actions on train operations by roaming adversarial drones.
police_officer_movements	Obstructions caused by stationary trains affecting movements of police officers.

H.4.4.3 Step 4.3) Define the assumptions made about factors

To do so, list all predicted emergent interactions without repetition. Then, describe the associated assumed situations, specifying which aspects of the factors are being assumed and which other aspects are not. Lateral Predictive Thinking Processes can be used to help with imagining those assumptions. For example,

Interaction 1: {roaming_adversarial_drone}_[-recognise]_{powered_powerlines_cables_

structure_visual_appearance}:

It is assumed that roaming adversarial drones are trained to recognise powered powerline cables based on their visual appearance. The trainer may not have considered random sparks or the frequency of powerline vibrations (wabbling).

Interaction 2: {moving_train}_[ignore]_{roaming_adversarial_drone}:

It is assumed that moving trains travel across clear train tracks in an orderly manner with no interruptions. However, moving trains unintendedly ignore roaming adversarial drones while approaching the train track zone. The adversarial drone has free and unobstructed access to the zone and is undetectable by any system.

For the entire table, see Appendix J, Table J.8. In this process version, we did not conduct any hazard analysis. However, in the following case study, we will incorporate hazard analysis to illustrate variations in the implementation and adaptability of safety analysis.

H.4.4.1 Step 4.4) Identify problematic Black Swan events

As we now have a full picture of most factors involved in the problem complexity, we can make an expert judgment and pick those that we deem rare events in such complexity and relative to the operational environment.

A. Current Concern Level
B. Rarely concerning
C. Impactful On solution ML model detection reliability

If cells are coloured green then the factor is a Black Swan with respect to the architect and the customer knowledge. The green factors are factors which are believed to be sources of surprise and potential Black Swan event may emerge by them or related to some aspect about them. We counted 41 potential sources of Black Swan events within the problem space.

Table H.13B Black Swan analysis

Predicted Factor	A	B	C
roaming_adversarial_drone	20.00%	no	yes
moving_train	17.85%	no	yes
train_tracks_structure	3.08%	no	yes
any_trained_computer_vision_agent_perception_capability	2.46%	no	yes
train_tracks_zone_fence_structure	2.46%	no	yes
train_derailment	2.15%	no	yes
fallen_vegetation_over_train_tracks_structure	2.15%	yes	yes
powered_powerlines_cables_structure_visual_appearance	1.85%	yes	yes
vegetation_complexity_growth	1.85%	yes	yes
police_officer_situational_awareness	1.85%	no	no

train_driver_situational_awareness	1.54%	no	no
available_ground_based_security_systems	1.54%	no	no
roaming_adversarial_drone_perception_capability	1.54%	no	no
roaming_adversarial_drone_flight_dynamics	1.54%	yes	yes
onboard_train_safety_systems	1.23%	no	no
extended_train_pantograph	1.23%	no	yes
train_tracks_structure_visual_appearance	1.23%	yes	yes
police_officer_capture_adversarial_drone	1.23%	no	yes
unsafe_train_tracks_zone_structure	0.92%	no	yes
unsafe_unsecured_passengers_and_goods	0.92%	no	no
alert_systems	0.92%	no	yes
roaming_adversarial_drones_location_data	0.92%	no	yes
weather_conditions	0.92%	no	yes
free_space_above_train_roof	0.92%	yes	yes
vegetation_complex_shapes_sizes_visual_appearance	0.92%	yes	yes
visible_powered_powerlines_cables_structure_visual_appearance	0.92%	yes	yes
police_force	0.92%	no	yes
stationary_train	0.92%	no	yes
unaware_train_driver_situational_awareness	0.62%	no	no
safe_passengers_and_goods	0.62%	no	yes
roaming_adversarial_drones_information	0.62%	no	no
unaware_emergency_response_teams	0.62%	no	no
onboard_train_communication_interface	0.62%	no	no
invisible_roaming_adversarial_drone_visual_appearance	0.62%	yes	yes
camouflage_mechanisms	0.62%	yes	yes
train_schedule_timing	0.62%	no	yes
train_tracks_maintenance_activities	0.62%	no	yes
track_zone_obstacles	0.62%	no	yes
visible_train_tracks_zone_fence_structure_visual_appearance	0.62%	yes	yes
passing_train_season_&_wind	0.62%	yes	yes
safe_secure_train_track_zone	0.62%	no	no
track_zone_monitoring_systems	0.31%	no	no
track_zone_monitoring_systems_data	0.31%	yes	yes
timely_track_zone_monitoring_systems_data	0.31%	yes	yes
accurate_track_zone_monitoring_systems_data	0.31%	no	yes
track_zone_monitoring_systems_data_gaps_or_inconsistencies	0.31%	no	yes

roaming_adversarial_drones	0.31%	no	yes
roaming_adversarial_drones_size_data	0.31%	yes	yes
roaming_adversarial_drones_flying_behaviour_data	0.31%	yes	yes
roaming_adversarial_drones_critical_data	0.31%	yes	yes
roaming_adversarial_drones_information_transmitted_data	0.31%	yes	yes
designated_emergency_response_communication_system	0.31%	no	no
emergency_notification	0.31%	no	no
reliable_railways_communication_network	0.31%	no	no
accurate_track_zone_monitoring_systems	0.31%	no	no
communication_network_connectivity	0.31%	no	no
alternative_communication_channels	0.31%	no	no
track_zone_monitoring_systems_data_plausibility	0.31%	yes	yes
ground_based_security_systems_received_data	0.31%	no	no
disruptive_adversarial_drones_threat_severity	0.31%	yes	yes
low_altitude_roaming_adversarial_drone	0.31%	no	yes
roaming_adversarial_drone_camouflage	0.31%	yes	yes
roaming_adversarial_drone_noise_emissions	0.31%	yes	yes
moving_train_blind_spots	0.31%	no	no
altitude_control_systems	0.31%	no	yes
roaming_adversarial_drone_noise_suppression_systems	0.31%	yes	yes
roaming_adversarial_drone_navigation_systems	0.31%	no	yes
altitude_parameters	0.31%	yes	yes
roaming_adversarial_drone_noise_output	0.31%	yes	yes
path_planning_algorithms	0.31%	no	yes
adversarial_drone_response_strategies	0.31%	yes	yes
flight_path_based_on_weather	0.31%	yes	yes
routes_around_maintenance_activities	0.31%	yes	yes
adversarial_drone_dangerous_navigation_around_obstacles	0.31%	yes	yes
random_vegetation_motion	0.31%	yes	yes
traveling_train_induced_air_turbulence_and_vibration	0.31%	yes	yes
fence_to_train_open_space	0.31%	yes	yes
train_electric_power_supply	0.31%	no	no
powered_powerlines_cables_structure_sparks	0.31%	yes	yes
visible_train_tracks_appearance	0.31%	yes	yes
visible_train_tracks_zone_fence_structure	0.31%	no	yes
visible_vegetation_visual_appearance	0.31%	yes	yes

visible_vegetation_structure	0.31%	yes	yes
powered_powerlines_cables_EMF	0.31%	yes	yes
powered_powerlines_cables_structure	0.31%	yes	yes
visibility_obstructive_objects	0.31%	yes	yes
adversarial_drone_collision_with_train	0.31%	no	yes
train_track_zone_structure	0.31%	no	yes
leaves_&_pollen	0.31%	yes	yes
passing_train_approach	0.31%	no	no
safe_secure_police_officer	0.31%	no	no
all_trains_operations	0.31%	no	no
police_officer_movements	0.31%	yes	yes

H.5 Stage 2: Architect Intent and Autonomous Solution Needs Definition[2]

This process would be used during Business or Mission Analysis in common systems engineering processes. It also defines the Operational Concept (OpsCon) for autonomous systems design. Although we didn't include the assessment of alternative solutions in this section, the output solution characterisation can guide the performance criteria to evaluate alternative solutions. This process also includes: **Stakeholder needs concept** definition on the back of solving problematic situations.

One pillar of AIC Systems Theory is the requisite of setting a purpose to realise streamlining the process in any system. The design team and their experiences, represented by the term "architect", is a system which requires an ideal purpose to aim for. The prime purpose of our approach is engineering an Ideal Whole, where an ideal autonomous systems solution can be realised or facilitated. Setting such a purpose sets the architect engineering approach in the right path towards making more comprehensive and objective design decisions, as it unifies the design team's ontology and language under a servitude of common PrimeP. This alone can add weight towards a more effective Trustworthiness Case for

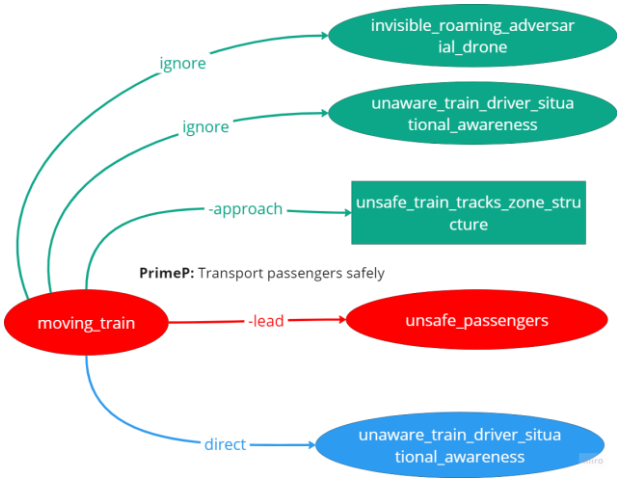
The architect's intent is also clearly defined at this stage. Architects define high-level objectives of what the autonomous systems should do and be validated to demonstrate. This is a collaborative stage between the architect and the wider stakeholders. A document is distributed among the stakeholders before a series of works HazTOPS dedicated to making decisions about the list of assumptions. Here is another design review gate where ethical considerations base

[2] See also section 6.3

decisions on what to do with the problem scope. In this PhD, ethical considerations processes have not been considered.

Autonomous systems' goals represent the high-level validation objectives the system is expected to fulfil. Those objectives can guide the safety case construction regarding the highest-level goals and how the design and test achieved those goals, including supporting artefacts. Below is an application of the above step:

Table H.15 Architect High-Level Solution Prescription

Situation	{moving_train}_[ignore]_{roaming_adversarial_drone} It is assumed that moving trains unintendedly ignore roaming adversarial drones while approaching the train track zone. 
Plausibility (Plausible/ Not plausible)	Plausible
Why?	Given the possibility of undetected adversarial drone intrusion into train tracks, moving trains are very likely to unintendedly approach unsafe train tracks, as if they are ignoring the fact that adversarial drones are already present within train tracks.
Architect Intent (mission)	Whole Solution PrimeP is to detect and physically neutralise adversarial schemes' impact on train operations across the train tracks zone. An intelligent security patrolling drone (Eagle Drone), supported by transport police officers, reports on train track zone safety.
Autonomous solution needs	• Restricted intelligent patrol train tracks zone within the boundary fence. • Intelligently detect drones within train track zone premises. • Inform ground control officers.

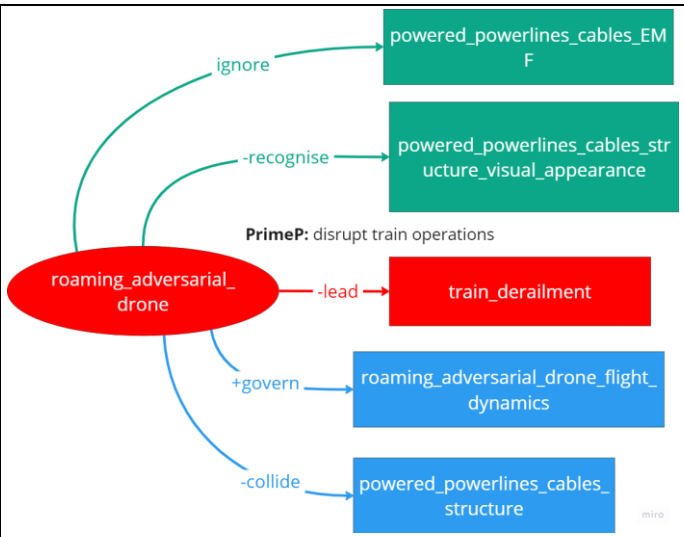
Autonomous solution constraints	• The autonomous patrol must remain within the defined perimeter of the train track zone and should not exceed the boundary fence to ensure that it does not interfere with external areas. • The autonomous system must maintain continuous surveillance coverage within the train track zone, ensuring no gaps in detection capability. • The patrol drone must accurately distinguish between authorised and unauthorised drones or objects within the train track zone. Misidentification must be minimised to prevent unnecessary alerts and optimise response coordination with transport police.
Support Systems Needs	Transport police officers analyse and manage feed from the Eagle Drone to respond appropriately.
Support Systems Constraints	While the Eagle Drone should function autonomously, ground control officers must be able to override its operation or manually control it during critical situations or emergencies to coordinate responses.

When the term “intelligent” is used in the requirement, it refers to intelligence as defined by autonomous systems. It refers to functionality that can be non-deterministic, with or without deterministic decision-making processes. For example, an intelligent detection method refers to a computer vision approach that combines machine learning algorithms (such as YOLOv8) and template matching methods. Intelligent patrolling refers to techniques that utilise deterministic means, such as automated algorithms, to follow predefined patrolling patterns, and non-deterministic means to decide, for example, which patrolling pattern to follow.

Let's have another example:

Table H.16 Architect High-Level Solution Prescription related to the impact of roaming adversarial drones

Situation	{roaming_adversarial_drone}_[-lead]_{train_derailment}. An assumption that roaming adversarial drones could potentially lead to train derailment.

	
Plausibility (Plausible/ Not plausible)	Plausible
Why?	Given the potential for adversarial drones to deliberately interfere with train operations, such as obstructing the train's path, damaging critical infrastructure, or distracting operational systems, the scenario of train derailment is plausible under these conditions.
Architect Intent (mission)	The Whole Solution PrimeP is to ensure train network operations remain uninterrupted by adversarial drones by detecting and neutralising threats effectively. The architect aims to use a combination of autonomous systems and human interventions to prevent derailment and maintain safety.
Autonomous solution needs	• The autonomous system should monitor and detect drones approaching train tracks and assess their threat levels. • Implement intelligent interception mechanisms to neutralise adversarial drones. • Provide live surveillance data and alerts to ground control for real-time decision-making.
Autonomous solution constraints	• Autonomous drones must avoid physical collisions with trains or track infrastructure. • Operations must be restricted to defined zones to avoid affecting neighbouring areas.
Support Systems Needs	• Ground control officers must receive real-time alerts and video feeds to make informed decisions. • Transport police officers need accurate situational reports to coordinate appropriate on-ground responses.

	• Integration with existing railway monitoring systems for enhanced situational awareness.
Support Systems Constraints	• Ground control officers must have override capabilities to manage critical situations. • Coordination protocols must be in place to prevent delays or errors in emergency response. • Supporting systems must maintain compatibility with autonomous drones and other railway infrastructure systems.

Table H.16 outlines the Architect High-Level Solution Prescription for addressing the risk of train derailment caused by roaming adversarial drones. The situation assumes that adversarial drones could intentionally interfere with train operations, leading to derailment. This scenario is deemed plausible due to the potential for drones to obstruct tracks, damage infrastructure, or distract operational systems. The architect's intent focuses on ensuring uninterrupted train network operations by effectively detecting and neutralising drone threats through a combination of autonomous systems and human interventions.

The autonomous solution needs emphasize real-time monitoring, threat assessment, interception mechanisms, and live data provision to ground control. To achieve this, the autonomous solution constraints mandate avoiding physical collisions, maintaining operations within defined zones, and ensuring the autonomous systems' alignment with surrounding infrastructure. Support systems needs highlight the importance of real-time alerts for ground control officers, situational reports for transport police, and seamless integration with existing railway systems. Additionally, support systems constraints require ground control override capabilities, robust coordination protocols, and compatibility with railway infrastructure to mitigate risks effectively. This comprehensive prescription addresses both operational and technical challenges, ensuring a proactive and integrated approach to enhancing train safety.

Table H.17 Architect High-Level Solution Prescription related to the police incapability to capture adversarial drone

Situation	|{police_officer_situational_awareness}_[+support]_{safe_secure_train_track_zone}| An assumption that Police Officers are incapable of catching adversarial drones.

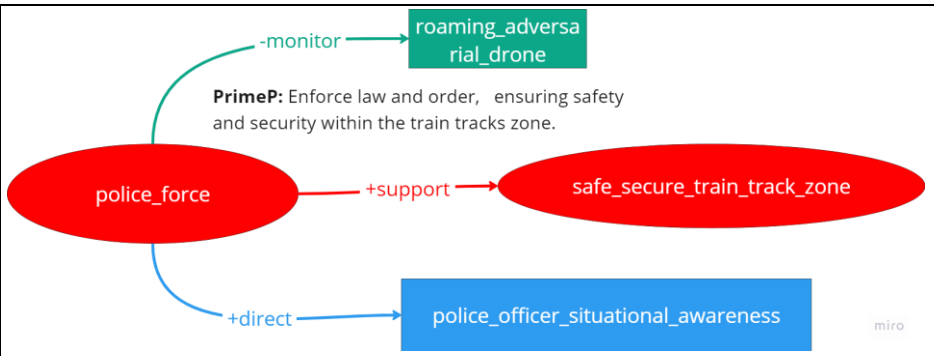
	
Plausibility (Plausible/ Not plausible)	Plausible
Why?	Adversarial drones' capabilities, including evasive manoeuvres and environmental adaptability, are more complex than police officers' physical and situational capabilities.
Architect Intent (mission)	The Whole Solution PrimeP is to enhance situational awareness and operational response capabilities of police officers to ensure safe and secure train track zones.
Autonomous solution needs	- Deploy intelligent autonomous drones to assist police officers by tracking and intercepting adversarial drones. - Provide real-time situational data to enhance decision-making. - Implement automated alert systems to inform officers of potential threats.
Autonomous solution constraints	- Autonomous drones must operate within predefined airspace boundaries. - Autonomous systems should avoid interfering with police personnel or operational infrastructure. - False positives and negatives in drone detection must be minimised.
Support Systems Needs	- Police officers require access to real-time drone tracking data and situational reports. - Advanced training programs for officers to collaborate with autonomous systems. - Integration with existing communication networks for seamless coordination.
Support Systems Constraints	- Systems must be user-friendly and require minimal training to ensure quick adoption. - Ground control officers must have override capabilities to manage critical situations. - Coordination protocols must prevent response delays during emergencies.

Table H.16 presents a comprehensive Architect High-Level Solution Prescription to address the incapability of police officers to effectively capture adversarial drones. The situation assumes that police officers lack the necessary capabilities to counter the advanced manoeuvring and adaptability of adversarial drones, a plausible scenario given the complexity of drone behaviours. The architect's intent focuses on enhancing police situational awareness and operational response through a combination of human and autonomous systems. The autonomous solution needs include deploying intelligent drones to assist in tracking and intercepting adversarial drones, providing real-time situational data, and implementing automated alert systems. To ensure operational efficiency, constraints require drones to operate within predefined boundaries, avoid interference with personnel, and minimise detection errors. Support systems needs emphasize real-time data access, training programs for officers to work collaboratively with autonomous systems, and integration with communication networks. Additionally, support systems constraints highlight the importance of user-friendly systems, override capabilities for ground control officers, and robust coordination protocols to prevent delays during emergencies, ensuring a secure train track zone.

H.6 Stage 3A: HazTOPS and Ordered AIC-driven Autonomous System Requirements Development[3]

This section will discuss hazards, threats and opportunities analysis when integrating the autonomous systems into the problem. For every defined Architect Prescription identified in stage 1, including the architect's intent, we model the chosen solution into the mix of interacting systems and re-evaluate the observed complexity. We need to model how the autonomous systems deals with every factor in the model.

H.6.1 Predictive Thinking Pipeline 1: Introducing Autonomous systems into Forward-Feed complexity

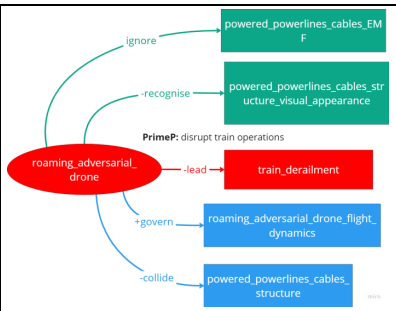
H.6.1.1 Step 1.1) Introduce the solution into the mix.

Model the AIC Schema and introduce the solution into the mix of the problem situation. We select the following interaction: {roaming_adversarial_drone}_[-lead]_{train_derailment}

Table H.18 Implementing Architect Intent and Forward-Feed AIC Interaction Framework for addressing train derailment caused by adversarial drones

Input	{roaming_adversarial_drone}_[-lead]_{train_derailment}

[3] See also section 6.4



Include Architect Intent (mission).

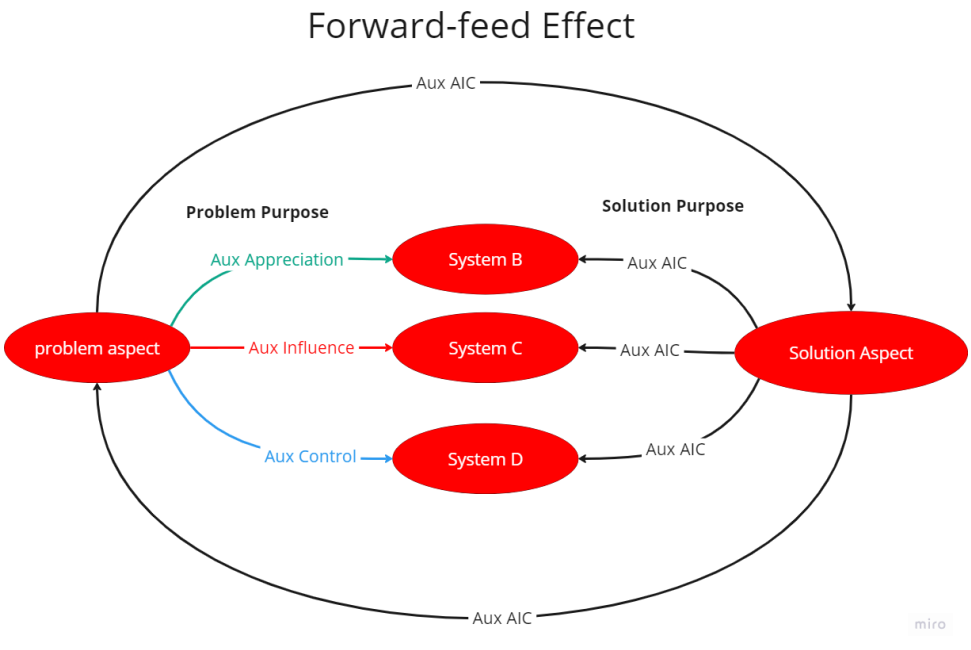
The Whole Solution PrimeP ensures that train network operations remain uninterrupted by adversarial drones by detecting and neutralising threats effectively. The architect aims to use autonomous systems and human interventions to prevent derailment and maintain safety.

- The autonomous system should monitor and detect drones approaching train tracks and assess their threat levels.
- Implement intelligent interception mechanisms to neutralise adversarial drones.
- Provide live surveillance data and alerts to ground control for real-time decision-making.

General Systems Rules

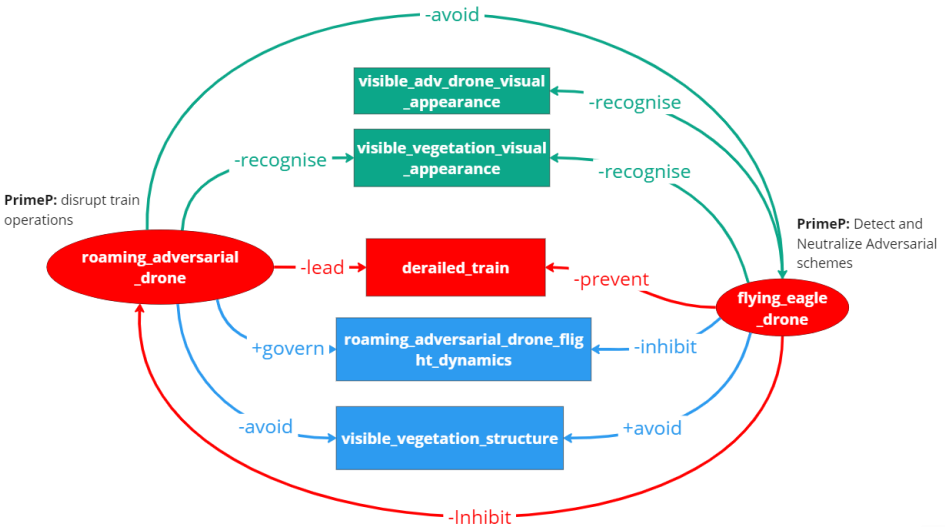
General rule G: Emergence of AIC Complicated behaviour.

General rule H: Forward-Feed Effect AIC modelling schema.



Predictive Thinking Process

Predictive question: What is the main problematic situation that needs to be influenced to achieve the influence? What situation needs to be controlled? What situation needs to be appreciated to ensure that the autonomous systems guarantee control?

	Guiding prompt: Define the autonomous systems Forward-Feed AIC interactions with the rest of the problem. Map the autonomous systems to all the parts of the problematic situation and define the actions and effect types for each interaction. 1. Start with the counter-influence intra-reaction to the main problematic situation. 2. Then, define which part of the problem needs to be controlled to achieve the influence. 3. Then, define which part of the problem needs to be appreciated such that the control can be achieved. **Step completion criteria:** The step is considered complete; All Forward-Feed AIC binary relationships have been modelled between the autonomous systems and the problematic situation.
Output Prediction	**Architect assertion:** The architect asserts that: 

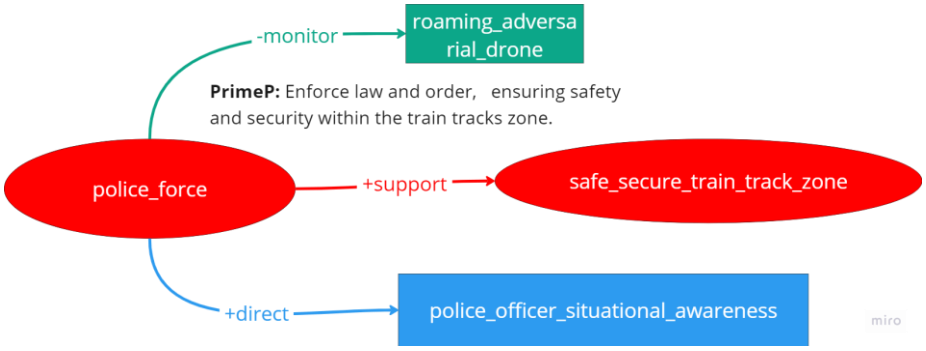
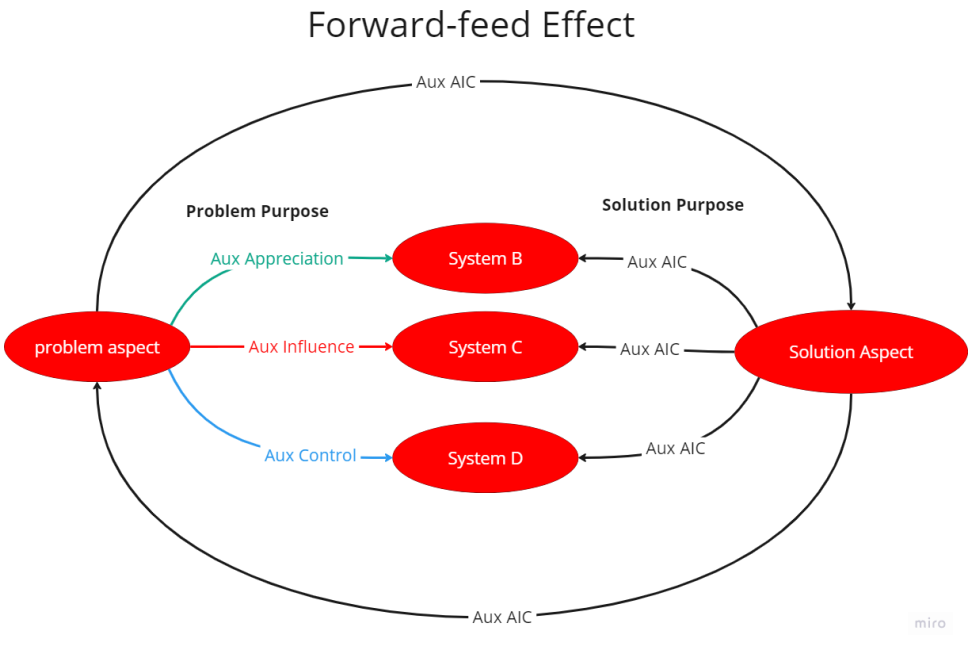
We identified an appreciative interaction between the roaming adversarial and Eagle Drones. This means we intend to re-engineer the complexity such that the adversarial drone shall never have any influence or control over the Eagle Drone and will always be influenceable by the Eagle Drone. This also means that autonomous systems interactions should facilitate such outcomes. The diagram stipulates that if the Eagle Drone:

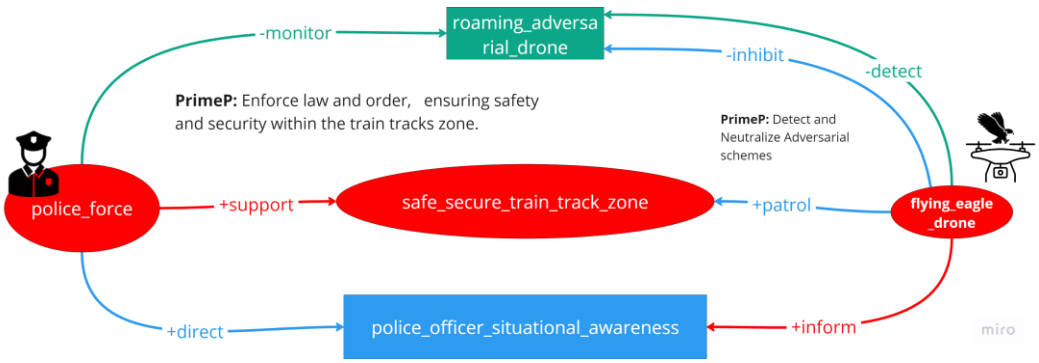
- Recognises the visibility of vegetation appearance.
- Avoids crashing into vegetation structures.
- Physically inhibit the by-passing drone.
- It cannot effectively be influenced or controlled by the adversarial drone.

Then we can sufficiently trust that the Eagle Drone can inhibit the adversarial drone and prevent derailment of the train incident.

Let's take another example,

Table H.18a a second example of deriving a scenario where the autonomous solution is introduced

Input	|{police_officer_situational_awareness}_[+support]_{safe_secure_train_track_zone}|  Include Architect Intent (mission). The Whole Solution PrimeP is to enhance situational awareness and operational response capabilities of police officers to ensure safe and secure train track zones. - Deploy intelligent autonomous drones to assist police officers by tracking and intercepting adversarial drones. - Provide real-time situational data to enhance decision-making. - Implement automated alert systems to inform officers of potential threats.
General Systems Rules	**General rule G:** Emergence of AIC Complicated behaviour. **General rule H:** Forward-Feed Effect AIC modelling schema. 
Predictive	**Predictive question:** What is the main problematic situation that needs to be influenced to achieve the influence? What situation needs to be controlled? What

Thinking Process	situation must be appreciated to ensure the autonomous systems guarantee control? **Guiding prompt:** Define the autonomous systems Forward-Feed AIC interactions with the rest of the problem. Map the autonomous systems to all the parts of the problematic situation and define the actions and effect types for each interaction. 1. Start with the counter-influence intra-reaction to the main problematic situation. 2. Then, define which part of the problem needs to be controlled to achieve the influence. 3. Then, define which part of the problem needs to be appreciated such that the control can be achieved. **Step completion criteria:** The step is considered complete; All Forward-Feed AIC binary relationships have been modelled between the autonomous systems and the problematic situation.
Output Prediction	**Architect assertion:** The architect asserts that: 

The image's AIC model represents a framework for addressing threats posed by roaming adversarial drones within a train track zone. It maps the interactions between key elements, police force, flying Eagle Drone, and the safe, secure train track zone, in relation to their respective roles and PrimeP.

- PrimeP Definitions:
  - The police force has the PrimeP to "enforce law and order, ensuring safety and security within the train tracks zone".
  - The flying Eagle Drone has the PrimeP to "detect and neutralize adversarial schemes".
- Appreciation:
  - The police force monitors the roaming adversarial drone indirectly through their situational awareness.

- The flying Eagle Drone detects the presence of the adversarial drone, appreciating its threat to the zone.

- Influence:
  - The police force supports the creation of a safe and secure train track zone through their operations and coordination.
  - The flying Eagle Drone informs the police force's situational awareness.
- Control:
  - The police officer situational awareness is directed by the police force to ensure informed actions.
  - The flying Eagle Drone directly inhibits the roaming adversarial drone's movements, maintaining control over potential threats.
  - The flying Eagle Drone patrols the zone, influencing the adversarial drone by neutralising its activities.

H.6.1.2 Step 1.2) Characterise the AIC interactions

Define the set of interactions between the sink and the source using AIC structured interaction format of: |{source situation}_[+,- or no sign, AIC-action]_{sink situation}|, Written in the following grammar: |{adjective+noun}_[verbal phrase]_{adjective+noun}| For example, |{flying_police_robot}_[learns humans' visual profiles]_{distressed_people}| Capture the output in the following table:

Table H.19 Mapping AIC interactions of the Eagle Drone and adversarial drone behaviours in mitigating train derailment risks

Source: {eagle _drone}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I1: |{flying_eagle_drone}_[+prevent]_ {derailed_train}|	**A1:** |{flying_eagle_drone}_[-recognise]_{visible_vegetation_visual_appearance}|
	C1: |{eagle _drone}_[-inhibit]_{roaming_ adversarial_drone_flight_dynamics}| **C2:** |{flying_eagle_drone}_[-avoid]_{visible_vegetation_structure}|

I2: |{flying_eagle _drone}_[-inhibit]_{roaming_ adversarial_drone}|	**A1:** |{flying_eagle_drone}_[-recognise]_{visible_vegetation_visual_appearance}| **A3:** |{flying_eagle_drone}_[-recognise]_{ visible_adv_drone_visual_appearance}|
	C1: |{flying_eagle _drone}_[-inhibit]_{roaming_ adversarial_drone_flight_dynamics}| **C2:** |{flying_eagle_drone}_[-avoid]_{visible_vegetation_structure}|
I3: |{roaming_adversarial_drone}_[-lead]_ {derailed_train}|	**A2:** |{roaming_adversarial_drone}_[-recognise]_ { visible_vegetation_visual_appearance }| **A3:** |{roaming_adversarial_drone}_[-avoid]_ {flying_eagle_drone}|
	C3: |{roaming_adversarial_drone}_[+govern]_ {roaming_adversarial_drone_flight_dynamics}| **C4:** |{roaming_adversarial_drone}_[+avoid]_ {visible_vegetation_visual_appearance}|

Table H.19 delineates the AIC interactions between the Eagle Drone and the roaming adversarial drone within the complexity field. It highlights how the Eagle Drone's behaviours, such as recognition, inhibition, and avoidance, counteract adversarial drone dynamics and mitigate risks to train operations, including derailment scenarios.

Table H.20 AIC structured interactions between the police force, flying Eagle Drone

Source: {police_force}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I4: |{police_force}_[+support]_ {safe_secure_train_track_zone}|	**A4:** |{police_force}_[-monitor]_ {visible_adv_drone_visual_appearance}| **A5:** |{police_force}_[-monitor]_ {roaming_ adversarial_drone_flight_dynamics}|

	C5: |{police_force}_[+direct]_ {police_officer_situational_awareness}| **C6:** |{police_force}_[+operate]_ {flying_eagle_drone}|
I5: |{flying_eagle _drone}_[+patrol]_{safe_secure_train_trac k_zone}|	**A3:** |{flying_eagle_drone}_[-recognise]_{ visible_adv_drone_visual_appearance}|
	C1: |{flying_eagle _drone}_[-inhibit]_{roaming_ adversarial_drone_flight_dynamics}| **C7:** |{flying_eagle _drone}_[+patrol]_{safe_secure_train_track_zone }|

H.20 field captures SECoT H.18a predicted output. While working on solving the police problem complexity field, we refined the initial model further. The following is the refined version of the initial complexity field in Figure H.7:

Output Behaviours (Influence):

- I4: The police force supports the creation of a safe and secure train track zone (|{police_force}_[+support]_{safe_secure_train_track_zone}|).
- I5: The flying Eagle Drone patrols the train track zone, directly influencing its safety (|{flying_eagle_drone}_[+patrol]_{safe_secure_train_track_zone}|).

A Behaviours (Appreciation):

- A4: The police force monitors the visible appearance of adversarial drones but with limitations (|{police_force}_[-monitor]_{visible_adv_drone_visual_appearance}|).
- A5: The police force also monitors adversarial drone flight dynamics but encounters challenges (|{police_force}_ [-monitor]_{roaming_adversarial_drone_flight_

  dynamics}|).

- A3: The flying Eagle Drone attempts to recognize adversarial drones visually but is not entirely effective (|{flying_eagle_drone}_[-recognise]_{visible_adv_drone_visual_

  appearance}|).

C Behaviours (Control):

- C5: The police force directs police officers' situational awareness to ensure safety and efficient action (|{police_force}_[+direct]_{police_officer_situational_awareness}|).
- C6: The police force operates the flying Eagle Drone, ensuring its deployment in patrolling and threat response (|{police_force}_[+operate]_{flying_eagle_drone}|).
- C1: The flying Eagle Drone actively inhibits the flight dynamics of roaming adversarial drones(|{flying_eagle_drone}_[-inhibit]_{roaming_adversarial_drone_flight_ dynamics}|).
- C7: The flying Eagle Drone performs patrols across the train track zone to maintain safety (|{flying_eagle_drone}_[+patrol]_{safe_secure_train_track_zone}|).

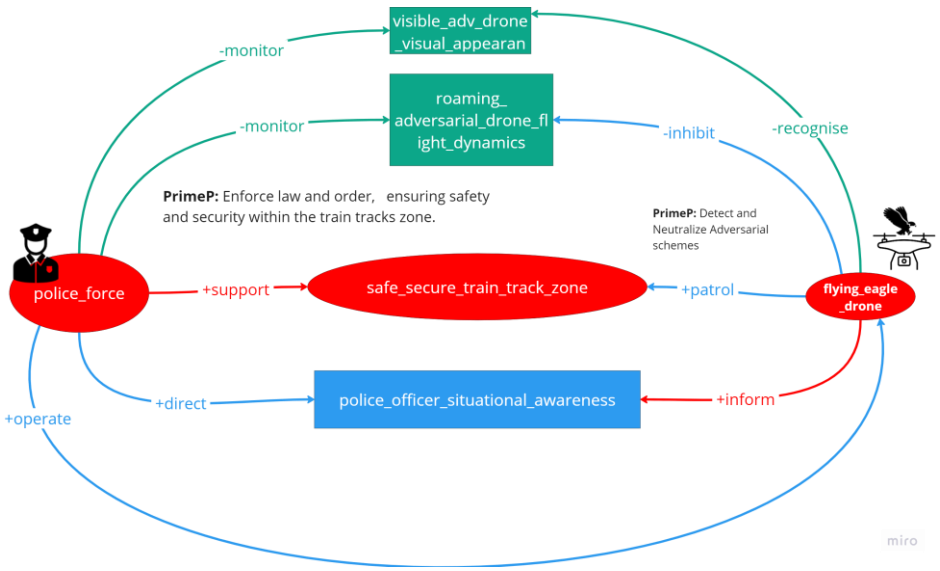


Figure H.6 Solution space complexity field for police force problematic interactions

H.6.2 Predictive Thinking Pipeline 2: Designing the affecting Backward-Feed complexity field

In this process, we will consider external factors that affect and are affected by the Complicated Behaviour. We will also consider systems that Appreciate, Influence, or control the Eagle Drone AIC behaviour.

H.6.2.1 Step 2.1) Visualise the operational design domain environment

Visualise the operational design domain environment with the Complicated Behaviour being part of it. To start modelling the operational environment, we need a real-world picture of the theatre of operations to help with specifying the operational environment complexity field.

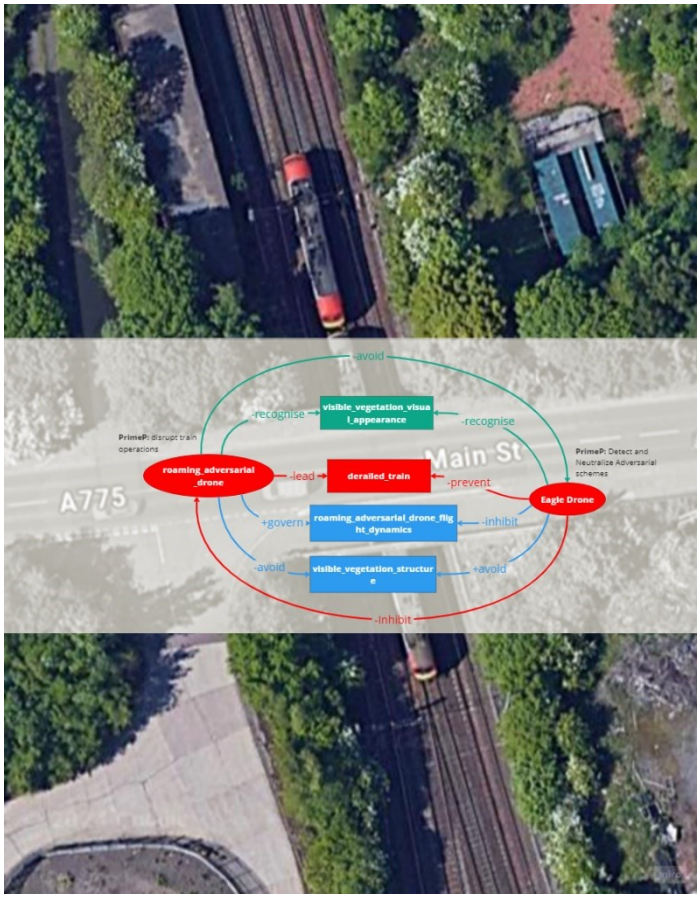


Figure H.7 A bird' eye view taken from the DuckDuckGo maps of the train track zone

From Figure H.8, we then produce the context diagram of the operational domain in Figure H.9. The model in Figure H.9 is designed to assist the architect in analysing its immediate operational environment and identifying potential risks and areas of interest in real-time. The model breaks down the environment into nine generic regions surrounding the Eagle Drone, categorising factors and elements that impact the drone's decision-making and operational capabilities. These regions provide context for object detection, tracking, avoidance, and interaction, enabling safe navigation and effective response to dynamic environmental conditions.

The model distinguishes between immediate factors directly affecting the Eagle Drone's operational decisions and remote factors indirectly impacting its operation. These two levels of influence are described below:

Remote Affecting & Affected Problem Complex

These elements indirectly influence the operational domain or are less likely to be directly encountered by the Eagle Drone. However, they can still affect its mission objectives and overall situational awareness.

- Police: The presence of law enforcement in the area may introduce additional operational constraints, such as no-fly zones or prioritised regions for surveillance. The Eagle Drone must know such remote factors to avoid conflict and comply with regulatory boundaries.

- Other Trains: Other trains in nearby but not directly adjacent tracks may influence the drone's flight path planning and situational awareness. Monitoring the movement of other trains can help the drone anticipate changes in its environment and avoid conflicts when operating near train tracks.

Immediate Affecting & Affected Problem Complex

These are elements that **directly impact or are impacted by the Eagle Drone's operations**. They represent the immediate environmental factors that the drone must continuously monitor, respond to, and navigate around.

- **Weather Conditions**: Real-time weather data is critical for flight safety, as conditions such as rain, wind, or fog can impair visibility and sensor performance. The Eagle Drone must adapt its path and speed based on current weather conditions.
- **Tracks North End / Tracks South End**: These represent the northern and southern ends of the train tracks in the drone's immediate operational area. The Eagle Drone monitors these areas to avoid flying too close to active train paths and to ensure safe operation around the tracks.
- **Active Birds**: Birds are a frequent and dynamic obstacle, posing a collision risk. The drone must detect, track, and avoid active birds to prevent accidents, especially in areas with high bird activity.
- **Disused Building**: This structure may present a potential obstacle or point of interest for surveillance. It may also affect signal reception and sensor accuracy, requiring the Eagle Drone to carefully navigate around it.
- **Clear Train Tracks**: Clear sections of train tracks allow for safe flight paths near ground transportation routes. However, if train traffic resumes, the drone may need to alter its path to avoid conflict.
- **Regions: North-East, North-West, South-East, and South-West**: These are geographical subdivisions of the operational domain, helping the drone to localize and respond to regional differences in environmental conditions and obstacles, such as:
    - **Oscillating Trees**: Trees swaying in the wind could obstruct the drone's vision and cause potential collision risks. The drone must account for these moving obstacles, especially in windier areas of the operational domain.
    - **Moving Train (South-East)**: An active train in the southeast region presents a moving obstacle that the Eagle Drone needs to track and avoid. Accurate positioning and velocity estimation are essential for safe operation around this moving hazard.

    - **Clear Pavement (South-West)**: This is a safe area where the drone might be able to hover or land if needed, as there are minimal obstructions and movement in this zone.

- **Roaming Adversarial Drone**: An adversarial or unauthorised drone in the area is a security risk. To maintain operational security, the Eagle Drone must identify, monitor, and avoid this potential threat.
- **Clear Road**: Sections of the road without vehicles provide safe low-altitude navigation zones. The drone may use these clear sections as part of its planned flight path to avoid traffic congestion.
- **Eastern / Western Over Bridges and Under Bridge**: These structural features can interfere with the drone's line of sight and sensor accuracy. The drone must be cautious when flying near or around these bridges, as they present collision risks and potential obstacles for flight planning.
- **Moving Automobiles**: Vehicles on nearby roads and bridges represent moving obstacles. The drone needs to track these vehicles to avoid potential collisions and plan safe, unobstructed paths.
- **Bridge Fence**: The fence along the bridge acts as a static obstacle that the drone must navigate around carefully.

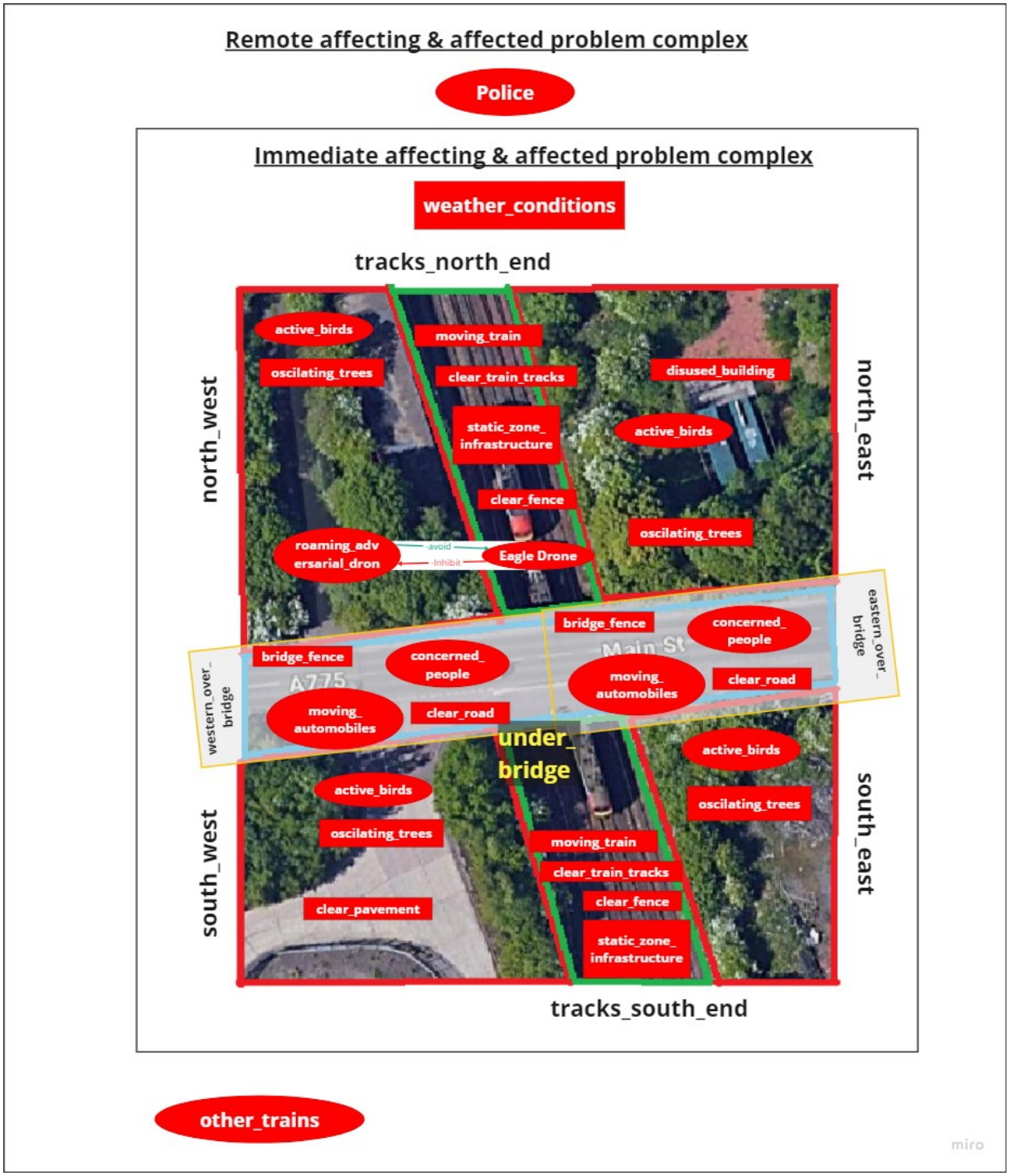


Figure H.8 Unsafe Train Tracks Operational Domain context diagram

The main idea here is to take a snapshot of the operational domain and start modelling the operational environment complexity field. H.9 is only an example of how we did it in our case study.

H.6.2.2 Step 2.2) Backward-Feed Complicated Behaviour definition

Extend the Forward-Feed AIC model to include the Backward-Feed Complicated Behaviour. In this step, we will extend the forward-feed model with the factors within the operational domain complexity by identifying which complexes are identified in the real-world problem complexity.

To do so, we need to abstract the Forward-Feed model for ease of modelling. We abstract the interaction model into the following definition:

Figure H.9 Abstract interaction for Eagle Drone denying adversarial drone access

Then, we model the Backward-Feed AIC interactions. We do so by re-interpreting and applying the backwards feed lateral Predictive Thinking Process described in section 5.7.4 & 7.7:

1. What complex appreciates the interaction? **We answer:**
    - Cars driving over the bridge can not influence or control the interaction.
2. What complex influences the interaction? **We answer:**
    - The visual dynamics of objects around the train tracks impact both Eagle Drone and Adversarial Drone. Perhaps, oscillating movements of trees could present a visual challenge for perception systems.
3. What complex has control over the interaction? **We answer:**
    - Police are in control of the interaction.

We will use icons to represent the complexes visually. Adding pictorial representations to the verbal description provides the designer with another dimension in visualising the scenario. Imagination is a key enabler for predictive thinking.

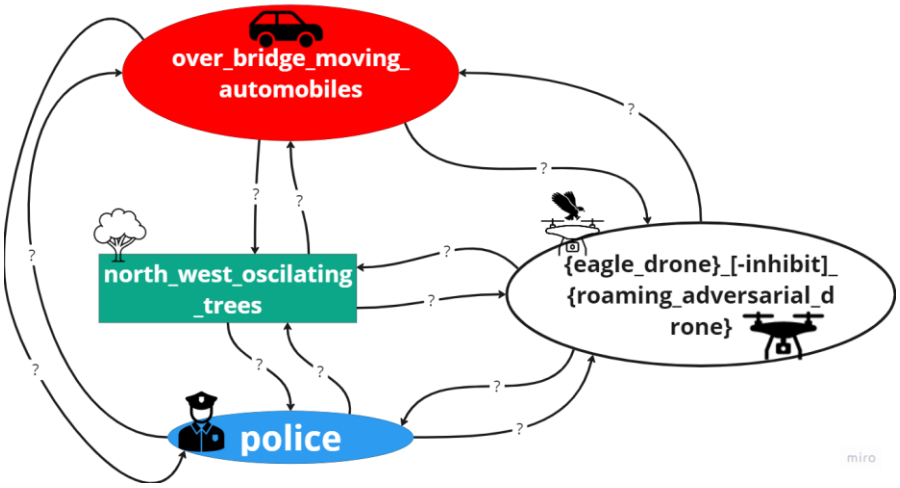


Figure H.10 Backward-feed complexity field

Looking at the operational domain context, we can see that the trees will impact both drones. Additionally, we recognise that police involvement is necessary, as cars driving over the bridge could also be impacted by a potential dogfight between two drones. Figure H.11 summarises

the backwards-feed complexity field. We needed to consider situations, not just complexes, so we had to define a special situation for each complex.

H.6.2.3 Step 2.3) Comprehensively appreciate the complicatedness of the problem domain

To comprehensively resolve the complicatedness of the above complexity abstraction, you may use the Actions Matrix method to define all the actions and their effect among the complexes. Note some actions do not make sense; for example, what would be the interaction between police and an oscillating tree? Here, lateral thinking is required to resolve this situation and articulate a plausible prediction. Also, note that the Eagle Drone may not have any intent to influence other complexes. Therefore, some of its actions are unsigned. The complexity involves the following complex of complexes:

Situation	Icon	Situation	Icon
flying_eagle_drone		oscillating_trees	
roaming_adversarial_drone		moving_automobiles	
police			

And the following complicatedness:

Table H.21 Actions Matrix 1

					
		-inhibit	+inform	avoid	distract
	-avoid		-avoid	gets in-between	Roam over
	+supervise	-capture		?	+ stop
	hinder	hide	?		?
	observe	observe	?	?	

Question marks denote a harder-to-predict (confusing) complicatedness since it is hard to comprehend how those complexes affect each other. The following is a list of hard-to-understand interactions:

- {police}_[?]_{oscillating_trees} and {oscillating_trees}_[?]_{police}
- {moving_automobiles}_[?]_{oscillating_trees} and {oscillating_trees}_[?]_{moving_automobiles}

The above interactions[4] make no sense, which indicates the limit of our predictive sequential thinking. To solve this complicated issue, we must think outside the box by applying lateral thinking. To do so, we may want to use the lateral thinking process in section F.9, where we rethink the interactions from different perspectives.

Key Interactions and Potential Scenarios:

1. Police and Oscillating Trees: {police}_[?]_{oscillating_trees} , {oscillating_trees}_[?]_{police}

Step 1: Assume a Broad Perspective Without Anchoring to Existing Rules

1.1 Acknowledge Unforeseen Impacts:

- Begin by considering that trees, although seemingly neutral, could have indirect or unforeseen impacts on drone operations. For example, they provide concealment for adversarial drones and potentially interfere with Eagle Drones' effectiveness by obstructing their line of sight.

1.2 Analyse Direct and Indirect AIC Interactions:

- **Direct Interaction (Control):** Tree removal enhances the Eagle Drones' ability to control the area by providing clear visibility.
- **Indirect Interaction (Appreciation/Influence):** Tree removal may influence adversarial drones to seek alternate concealment strategies or affect the community's appreciation of police actions, potentially leading to social resistance.

Step 2: Generate Analogies Across Different Contexts

2.1 Draw Analogies:

[4] Interestingly, a 5X5 matrix renders 25 interactions, Log(25,2) = 4.6, and we had 5 unknown unknowns. This is a close enough approximation. This tells us some interesting aspects about interpreting the epistemic uncertainty value Log(N). This could mean the likelihood of finding a minimum number of unknown unknowns in a set of things. Hence, it is reasonable to speculate that residual ignorance modelling based on Log(N) gives us an insight into how much knowledge we may be missing. However, this is a matter for future research.

- **Urban Planning Analogy:** Removing trees to clear line-of-sight is similar to removing barriers to reduce blind spots in urban surveillance systems. However, this can lead to unintended consequences, such as altering traffic patterns or reducing public trust.
- **Military Tactics Analogy:** Like camouflage nets in military settings, trees act as natural camouflage for adversarial drones, complicating detection.

2.2 Explore Counterintuitive Scenarios:

- Imagine a humorous yet revealing situation: “It would be hilarious if removing trees for better visibility accidentally led to drones getting tangled in exposed overhead wires, causing operational failures”.
- Another scenario could involve unexpected community pushback: “What if locals, upset about the tree removal, started planting fake trees as a form of protest, confusing the Eagle Drones further?”

Step 3: Formulate Hypotheses and Generate New Operational Scenarios

3.1 Hypothesize New Scenarios for System Recognition:

- **Scenario 1:** "Drones navigating areas with dense tree cover and shifting light patterns caused by oscillating branches”.
- **Scenario 2:** "Operational risks and social resistance resulting from large-scale tree removal efforts”.
- **Scenario 3:** "Adversarial drones deploying tree-mimicking camouflage to exploit community-planted vegetation as new concealment zones”.

2. Moving Automobiles and Oscillating Trees ({moving_automobiles}_[?]_{oscillating_trees}), ({oscillating_trees}_[?]_{moving_automobiles})

Step 1: Assume a Broad Perspective Without Anchoring to Existing Rules

Thought Step 1.1: Define the Context Without Fixed Assumptions

- Begin by considering the environment where moving automobiles and oscillating trees coexist. Avoid the assumption that trees and vehicles are unrelated or independent elements. Instead, acknowledge that both moving automobiles and oscillating trees could interact in unforeseen ways to influence drone operations.
- Example: Oscillating trees may block or distort the line of sight for drones while moving automobiles introduce dynamic obstacles that challenge drone agility and decision-making.

Thought Step 1.2: Consider Direct and Indirect Interactions

- **Direct Interaction (Control):** Oscillating trees could directly obstruct the path of a drone during pursuit or surveillance, potentially leading to collisions.
- **Indirect Interaction (Appreciation/Influence):** The motion of large Vehicles (lorries or military vehicles) could create turbulence or alter air pressure, influencing how trees oscillate, which in turn affects drone stability or navigation accuracy.

Step 2: Generate Analogies Across Different Contexts

Thought Step 2.1: Draw Analogies to Explore Influence Patterns

- **Wind and Kites Analogy:** Oscillating trees can be likened to wind influencing the flight of kites. Just as unpredictable gusts can destabilise a kite, tree movement can disrupt drone perception stability during operations near moving vehicles.
- **Traffic and Pedestrian Analogy:** Like pedestrians navigating through moving traffic, drones must simultaneously avoid both static (trees) and dynamic (vehicles) obstacles.

Thought Step 2.2: Explore Counterintuitive or Surprising Scenarios

- Imagine a surprising situation:
    - **Humorous Scenario:** “It would be hilarious if a drone mistook the oscillating branches of a tree for the moving arms of a pedestrian, causing it to track the tree instead of focusing on an actual intruder”, assuming the Eagle Drone is designed to rack pedestrians.
    - Another counterintuitive situation could involve a gust of wind caused by a passing truck amplifying tree oscillation, leading to a chain reaction where a drone miscalculates its trajectory and collides with a nearby powerlines.

Step 3: Formulate Hypotheses and Generate New Operational Scenarios

Thought Step 3.1: Hypothesise New Scenarios and Patterns for Recognition

- **Scenario 1:** "Dynamic interaction between oscillating trees and vehicles in windy conditions causing drones to miscalculate navigation paths”.
- **Scenario 2:** "Tree movements mimicking human or animal behaviours, leading to false positives in drone threat detection systems”.
- **Scenario 3:** "Combined influence of moving automobiles and oscillating trees creating turbulence that destabilises drone flight paths, increasing collision risks with safety-critical structures such as power-lines"

Black Swan Factors: agitated_people and **sparking_powerlines**

Agitated People ({agitated_people})

- Local resistance could arise if police take measures that affect environmental elements valued by the community, such as trees, birds, or local green spaces. Actions perceived as environmentally disruptive may lead to public unrest, complicating police operations, and potentially decreasing local support for drone surveillance efforts.
- This factor could influence policy or regulatory discussions, where the community demands more environmentally friendly approaches to security, such as alternative placement for drones or less invasive vegetation management techniques.

Sparking Powerlines ({sparking_powerlines})

- This factor addresses the potential consequences of drone dogfights or falling trees on power infrastructure. Drones, when flying low or engaging adversarial drones, might come close to powerlines, and if a collision occurs, short circuits could spark during dry conditions, causing fires and potentially damaging the electrical grid.
- This risk may necessitate stricter flight paths or protocols for police drones to avoid low-hover zones near powerlines or windy areas. Moreover, community members may voice safety concerns, prompting both police and power companies to work together to mitigate these hazards.

Iterative thought: in section H.1.2.3, which is a step related to design problem selection, we initially considered the following design decision:

interaction	Potential concern	Decision	Elaboration or Justification
n26	Connected train pantograph to electric powerlines (sparks and wobble). This may impede the ability to perceive trained computer vision agents.	To be dropped	Although it is prevalent, sparks are unlikely to be generated often enough for us to include it as a design concern.

We decided to drop, considering power sparks as a significant factor. In this section, we realised that, actually, although they are rare, they can be very impactful. It is, in fact, a Black Swan event.

Therefore, we decided to change our minds and include it as part of the problem domain. This means we will need to update stage 2. It is rather convenient to come up with this realisation at this stage than only to realise it after deployment or even after we finalise or baseline system-level requirements for the design.

Below are the updated classes of systems involved in the complexity field:

Situation	Icon	Situation	Icon
flying_eagle_drone		moving_automobiles	
police		agitated_people	
oscillating_trees		sparking_powerlines	

Thus, we can extend Action Matrix 1 in Table H.20:

Table H.22 Action Matrix 1.1

						
		+inform	avoid	fly over	-threaten	avoid
	+supervise		-remove	+stop	-complicate	prevent
	hinder	- complicate		complicate	comfort	-complicate
	ignore	- complicate	-remove		ignore	observe
	-interfere	-obstruct	+protect	ignore		avoid
	Interfere	obstruct	burn	Interfere	deter	

H.6.2.4 Step 2.4) Complete the extended view of the AIC mental model schema

(a) Roaming adversarial drone problem solution space complexity field:

The Actions Matrix above defines how the complexity of the scenario resolves. Now, we will transfer all the knowledge to the final output of the step, which is a comprehensive AIC mental model of the problem. Note how we started and how we ended. Every step in this process carefully explains how we produced the scenario. you will notice that we change the AIC relationship in some interactions, for example, the relationship between  and  . Initially, we imagined that the action “collide” is an influence-type action with an influence goal however as we visually modelled the AIC interaction, we discovered it is more accurate to define the

collision as a control-based interaction rather than influence since the drone is directly impacting the powerlines through collision. We then capture all the behaviours modelled with the matrix in an AIC Complexity Field model.

Note that the interactions between the Forward-Feed bubble and the Backward-Feed operational environment concern the source. At this moment in the design, we are interested in the Eagle Drone operation. Later, if the design team wishes to evaluate the complexity from the sink perspective, the complexity will change. The approach is flexible and allows for investigating and modelling the whole from any node, thus creating a variety of complexities.

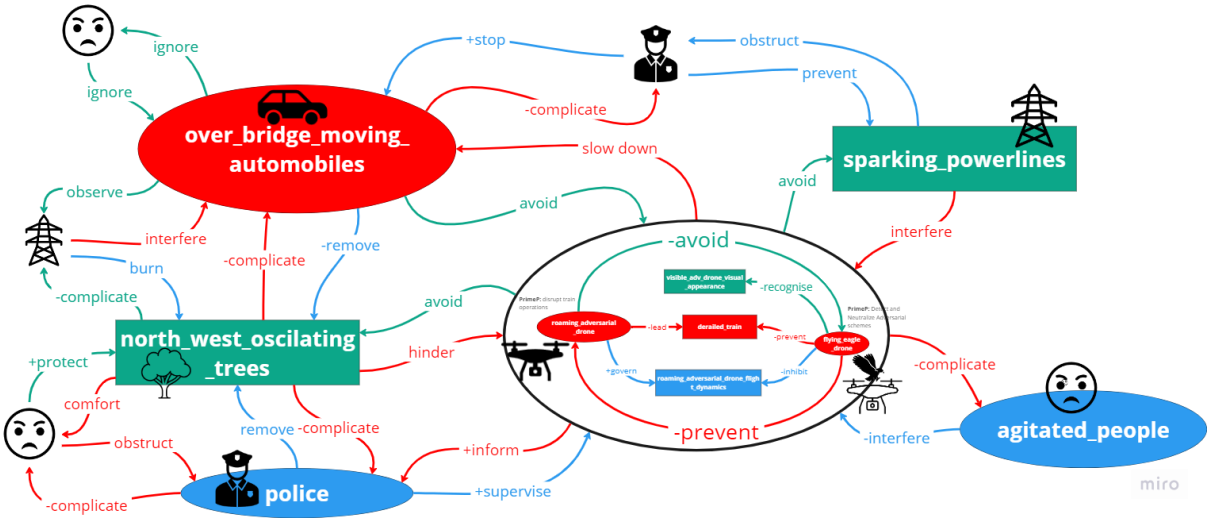


Figure H.11 AIC Complexity Field Eagle Drone interaction with adversarial drone

(b) Police force problem solution space complexity field:

Let's take another example; in this case, we will consider the complexity field modelled in Figure H.7. In Figure H.13, we used the operational environment complexity field.

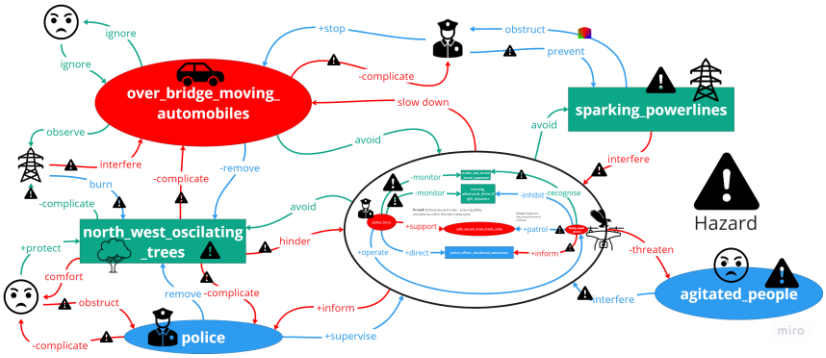


Figure H.12 Police force problem solution space complexity field

H.6.2.5 Step 2.5) Capture the AIC interactions

When looking at the complexity from any complex's point of view, the main objective is to examine how the rest of the complexity affects the complex's chances (probability) of manifesting its PrimeP. For example, let's evaluate the complexity of an adversarial drone's ability to avoid the

Eagle Drone. We would determine the AIC complexity and how other complexes impact the chances of the adversarial drone manifesting its PrimeP. Such analysis may allow us to determine the circumstances and factors involved with adversarial drone operations, thus giving us more insight into the potential emergence we must deal with. This step captures each node source and defines the AIC behaviours. In this case, we will start with our system of interest, the Eagle Drone and extend its original AIC behaviours.

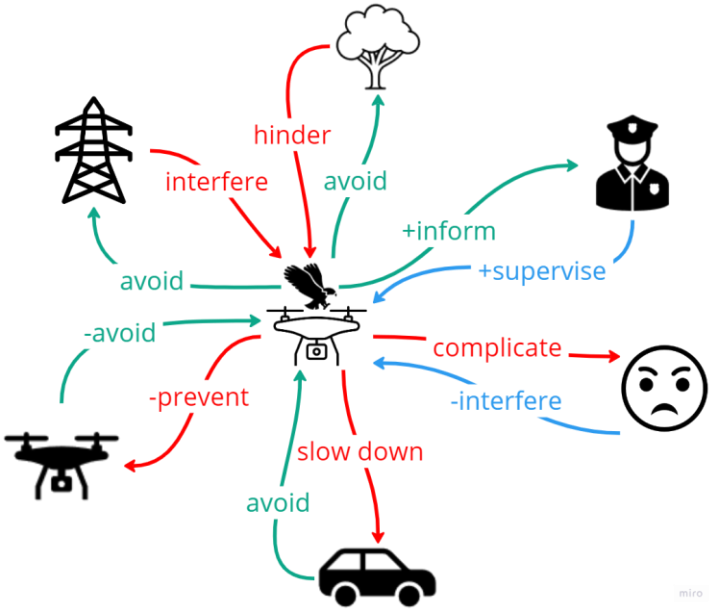


Figure H.13 Eagle Drone Complexity Field

Note in the table below, C5, A6, A7, as well as the crossed parameters, signify an update of initial inaccurate or unknown situations to a more accurate understanding of the overall complicatedness.

Table H.23 The table outlines the interactions and behavioural influences of Eagle Drone

Source or Sink: {eagle _drone}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I1: |{flying_eagle_drone}_[+prevent]_ {derailed_train}|	**A1:** |{flying_eagle_drone}_[-recognise]_{visible_adv_drone_visual_appearance}|
	C1: |{eagle _drone}_[-inhibit]_{roaming_ adversarial_drone_flight_dynamics}|
I2: |{flying_eagle _drone}_[-prevent]_{roaming_ adversarial_drone}|	**A1:** |{flying_eagle_drone}_[-recognise]_{visible_adv_drone_visual_appearance}|

	C1: |{eagle _drone}_[-inhibit]_{roaming_ adversarial_drone_flight_dynamics}|
I3: |{roaming_adversarial_drone}_[-lead]_ {derailed_train}|	**A2:** |{roaming_adversarial_drone}_[-avoid]_ {flying_eagle_drone}|
	C2: |{roaming_adversarial_drone}_[+govern]_ {roaming_adversarial_drone_flight_dynamics}|
I4: |{flying_eagle_drone}_[-complicate]_ {agitated_people}|	**A6:** |{flying_eagle_drone}_[+avoid]_ {visible_north_west_oscilating_trees}| **A7:** |{flying_eagle_drone}_[+avoid]_ {visible_train_tracks_structures}|
	C3: |{agitated_people}_[-interfer]_ {flying_eagle_drone}| **C5:** |{agitated_people}_[-interfer]_ {flying_eagle_drone}|
I5: |{flying_eagle_drone}_[slow_down]_ {over_bridge_moving_automobiles}|	**A3:** |{over_bridge_moving_ automobiles}_[avoid]_{flying_eagle_drone}|
	To be defined
I5: |{flying_eagle_drone}_[+inform]_ {police}|	**To be defined**
	C4: |{police}_[+supervise]_ {flying_eagle_drone}|
I6: |{sparking_powerlines}_[interfere]_ {flying_drone}|	**A4:** |{flying_eagle_drone}_[avoid]_ {sparking_powerlines}|

	To be defined
I7: |{north_west_oscilating_trees}_[hinder] _{flying_drone}|	**A5:** |{flying_eagle_drone}_[avoid]_ {north_west_oscilating_trees}|
	To be defined

Note that there are several unresolved complicatedness named “to be defined”. This is where further AIC modelling uncovers those intricate interactions. Also, note that the relationship between the Eagle Drone and police was thought of initially to be appreciation; however, while we were capturing the interactions in the controlled format, we realised that it is better to consider it as an influence since the Eagle Drone does influence police behaviour. Below is a holistic overview of the behaviours of Eagle Drones in their operational environment. Note that we changed the colour of the interaction between the Eagle Drone and the people to indicate that the influence interaction has been resolved. While the remaining influence interactions remain in red, which means they are yet to be resolved.

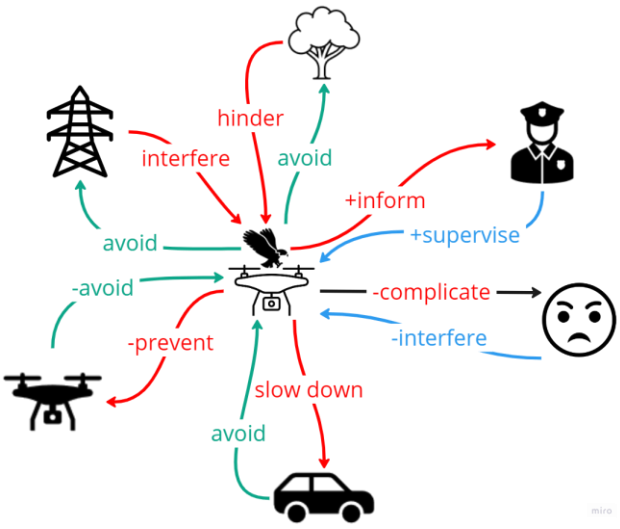


Figure H.14 Corrected Eagle Drone complexity field

H.6.3 Predictive Thinking Pipeline 3: Hazards, Threats and Opportunities Scenarios (HazTOPS) Analysis

We apply the HazTOPS SECoT outlined in section [reference]. The following is the application of the process:

H.6.3.1 Step 1) Scope the HazTOPS context domain

Scope the potential safety and security challenges on the AIC schema of the problem domain.

Add the following icon ⚠ for safety hazards, 🏅 for opportunities and 🥷 for potential security threats (cyber-attacks) to every interaction on the mode.

Note that the *n* interaction between agitated_people and flying_eagle_drone has changed from complicated to “threaten” this is because when we produced the HazTOPS for this interaction, we discovered that the complicated interaction is “threatening”.

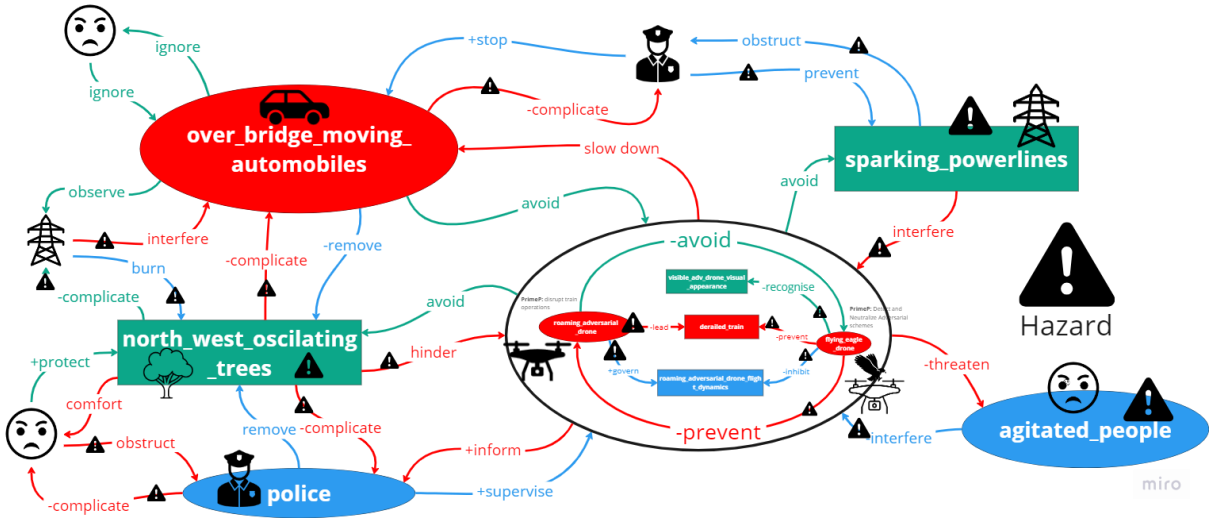


Figure H.15 Hazards Complexity Field Scope: graphically scoping the hazards within the complexity field by placing hazard icons on target interaction.

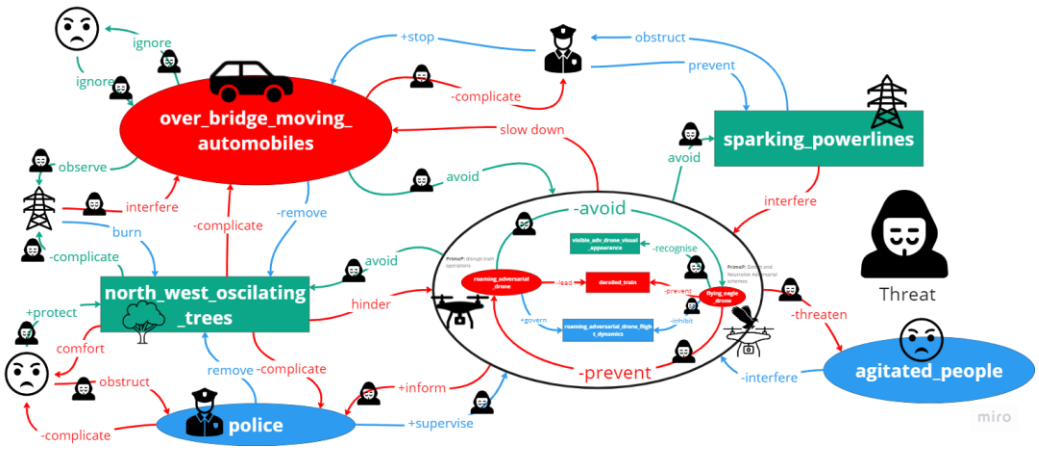


Figure H.16 Threats Complexity Field Scope

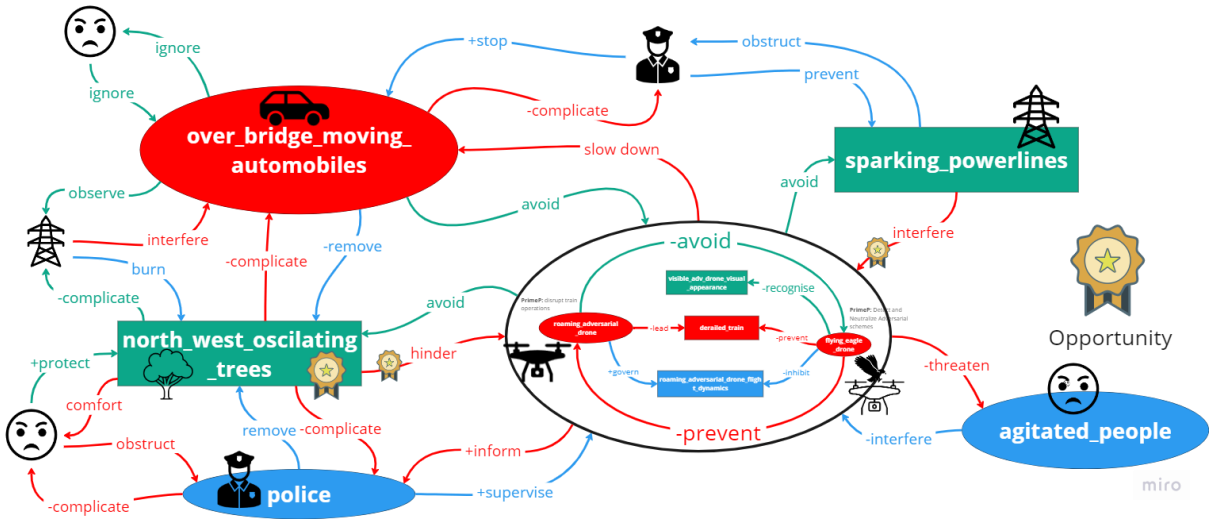


Figure H.17 Opportunities Complexity Feels Scope

The opportunities identified in the Complexity field are the following:

- Trees can also hinder adversarial drone operations, thus providing natural deterrence. Knowing this, the police may decide to keep the trees rather than remove them or even increase their density, as it is a cheaper deterrent.
- Sparking powerlines can also interfere with adversarial drone operations.

Therefore, we can conclude that the train track zone is not totally defenceless. There are natural defences that can be leveraged and included in the overall strategy, thus reducing the overall solution's costs.

H.6.3.2 Step 2) Characterise the scoped interactions.

H.6.3.3 Example 1: Roaming adversarial drone

Clearly define the nature of the scoped interactions. We will use only the intelligent system's Forward-Feed model in this step. However, the process requires considering all interactions in the complexity field.

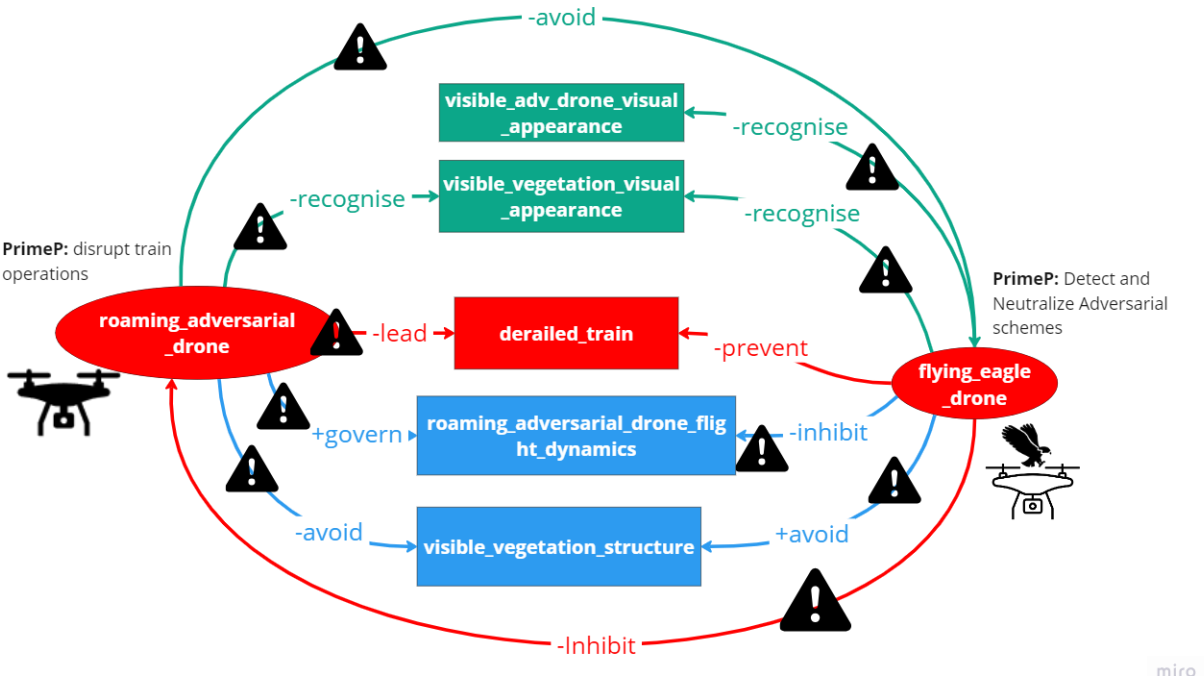


Figure H.18 Hazards associated with Eagle Drone preventing derailed train complexity field

Below is the AIC interactions definition table of the above model:

Table H.23 illustrates the behaviours and interactions within the AIC model, focusing on Eagle Drones as the source. It categorises specific behaviours (Output Behaviours) of the Eagle Drone, such as inhibiting adversarial drones and their associated contributing (AC) behaviours. These include recognition of visual appearances, avoidance of environmental structures like vegetation, and flight dynamics inhibition. The table emphasises the intricate dependencies and hazard mitigation strategies within surveillance operations.

Table H.24 The table describes the AIC interaction dynamics between Eagle Drones and adversarial drones

Source: {eagle _drone}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I2: |{flying_eagle _drone}_[-inhibit]_{roaming_ adversarial_drone}|	**A1:** |{flying_eagle_drone}_[-recognise]_{visible_vegetation_visual_appearance}|

	A3: |{flying_eagle_drone}_[-recognise]_{ visible_adv_drone_visual_appearance}|
	C1: |{eagle _drone}_[-inhibit]_{roaming_ adversarial_drone_flight_dynamics}| **C2:** |{flying_eagle_drone}_[-avoid]_{visible_vegetation_structure}|

H.6.3.4 Step 3) Apply predictive potential complications guide words.

Then, identify further potential complexity by utilising the following modified keywords: More, Part of, Less, Early, and Late. Follow SECoT_2 to derive the variety of potential deviations. Then include a risk and surprise analysis for each HazTOPS scenario.

Table H.25 HazTOPS Analysis of "3 Drones Attack" Scenario

Title	3 drones attack
AIC interaction	**I2:** |{flying_eagle _drone}_[-inhibit]_{roaming_ adversarial_drone}|
HazTOPS Aspect	**Definition**
Hazards, Threats or Opportunities Scenario: Guide word	More: 3 adversarial drones involved.
Operating Scenario Context	Clear day, and the Eagle Drone is conducting a patrol.
Hazard, Threat or Opportunity definition (consider used-systems)	More than 1 adversarial drone approach.
Foreseeable Sequence of Events	• The Eagle Drone is hovering over a designated 360-degree panoramic survey. • While facing the fence direction, three adversarial drones appear within 10 meters of the Eagle Drone. • The Eagle Drone correctly identify three adversarial drones.

	• It decides to contact HQ. • It launches pursuit and inhibits on one of the drones. • It crosses a boundary fence. • The aerial dog fight occurs within the track zone.
Potential harm or benefit	• Potential crash on track zone infrastructure. • Potential ingression over local houses' back gardens. • The other drones are freely roaming around train track zone.

Table H.25 provides an analysis of the "3 Drones Attack" scenario using the HazTOPS framework, focusing on risks, threats, and opportunities in a situation where an Eagle Drone must manage multiple adversarial drones. The AIC interaction highlights the Eagle Drone's objective to inhibit roaming adversarial drones, while the HazTOPS Aspect introduces the guideword "More," signifying an increased threat with three adversarial drones involved. In the operating scenario context, the event occurs on a clear day during a routine patrol. The foreseeable sequence of events outlines the Eagle Drone's identification and response to the threat, including contacting HQ, launching a pursuit, engaging in an aerial dogfight, and even crossing a boundary fence. The analysis identifies potential harms, such as crashes on track zone infrastructure, encroachments into private properties, and the inability to manage all drones simultaneously, leaving some adversarial drones to roam freely within the train track zone. This assessment emphasises the complexity and potential risks of managing multiple simultaneous threats.

Table H.25 presents a HazTOPS analysis of the scenario in which adversarial drones use smart lasers to target the Eagle Drone. The AIC interaction describes the Eagle Drone's role in recognising the adversarial drone's visual appearance. The HazTOPS Aspect focuses on the guideword "Part of," highlighting that a specific part of the Eagle Drone, the camera system, is deliberately targeted. In the operating scenario context, the Eagle Drone operates under "Grade A" environmental difficulty during a routine patrol, emphasising the operational baseline and normal conditions for this event.

The hazard or threat definition specifies the adversarial drone’s capabilities, which include a high-energy laser system paired with an intelligent tracking computer. This setup allows the adversarial drone to track and target the Eagle Drone's camera system with precision. The foreseeable sequence of events details how the Eagle Drone conducts surveillance along the track zone fence, stopping for a 360-degree view. During this time, the adversarial drone,

positioned at a safe distance, uses its laser system to blind and disable the Eagle Drone's camera system, evading detection in the process.

Table H.26 HazTOPS Analysis of adversarial drone using smart lasers scenario

Title	Adversarial drones use smart lasers
AIC interaction	**A3:** |{flying_eagle_drone}_[-recognise]_{ visible_adv_drone_visual_appearance}|
HazTOPS Aspect	**Definition**
Hazards, Threats or Opportunities Scenario: Guide word	Part of: Part of the Eagle Drone is specifically targeted.
Operating Scenario Context	Operational environment difficulty grade A. Eagle Drone conducting a usual patrol.
Hazard, Threat or Opportunity definition (consider used-systems)	The adversarial drone has an energy laser system and an intelligent, dedicated computer to track and target the Eagle Drone's camera. The aim is to damage the Eagle Drone camera physically.
Foreseeable Sequence of Events	The Eagle Drones fly along the track zone fence and stop for 360-degree surveillance. The adversarial drone, from a long enough distance, targets the Eagle Drone camera system with a high-energy laser beam while approaching the train track zone. The laser system successfully blinds the Eagle Drone camera and evade detection.

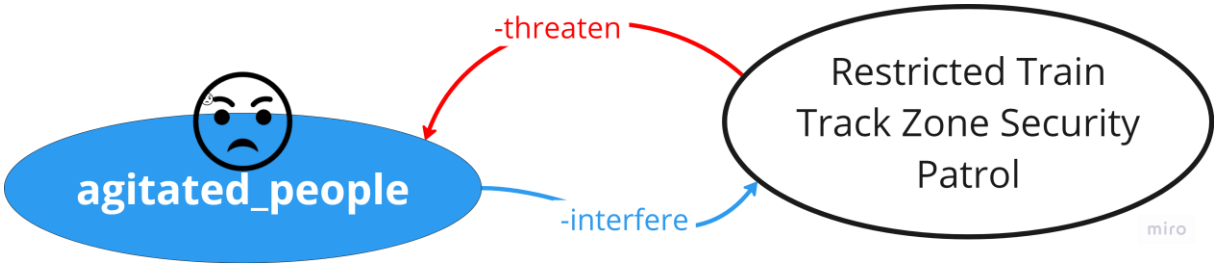
Table H.26 presents a HazTOPS analysis of a scenario where an adversarial drone hides behind a train track fence to evade detection. The AIC interaction describes the Eagle Drone's inability to recognise the visual presence of the adversarial drone, which is less visible due to its strategic position. The HazTOPS Aspect uses the guideword "Less," highlighting reduced visibility as a critical vulnerability. The operating scenario context indicates a challenging operational environment (difficulty grade A) and notes that the area adjacent to the fence is not included in the Eagle Drone's training data, further complicating detection. The hazard definition describes how the adversarial drone hides behind the fence, evading detection until it launches an attack as a train passes. The foreseeable sequence of events outlines the drone's stealth approach, leading to a failure by the Eagle Drone to respond in time, potentially resulting in train derailment, illustrating the severe risks posed by this scenario.

Table H.27 HazTOPS Analysis of adversarial drone hiding behind fence scenario

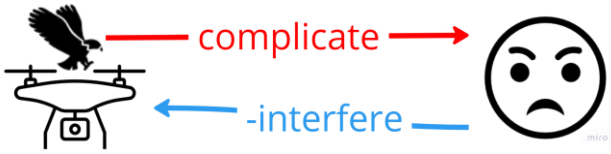
Title	Adversarial drone behind fence
AIC interaction	**A3:** |{flying_eagle_drone}_[-recognise]_{ visible_adv_drone_visual_appearance}|
HazTOPS Aspect	**Definition**
Hazards, Threats or Opportunities Scenario: Guide word	Less: Adversarial drones are less visible
Operating Scenario Context	Operational environment difficulty grade A. Eagle Drone conducting a usual patrol. The area adjacent to the train track zone fence is not part of Eagle Drone training data.
Hazard, Threat or Opportunity definition (consider used-systems)	Adversarial drones hide behind train track fence
Foreseeable Sequence of Events	The adversarial drone cunningly lands on the floor behind the fence; the Eagle Drone does not notice its presence; train passes and the adversarial drone launches to strike the train; the Eagle Drone fails to act swift enough.
Potential harm or benefit	Train derailment

H.6.3.5 Example 2: Police Force problem

Let's take a different relationship; this time, we will consider the police force solution space complexity field.



One of the affected situations is also people; however, this time, the main threat is infringing on their privacy:



First, we need to interpret this AIC relationship. The Eagle Drone complicates the situation of agitated people. Note that the action has no sign, which means it is unintended, while the agitated people's action is intended to be obstructive. During the process, we realised that the reason behind agitated people's feelings is the sense of threat. It is more appropriate to assume the influence action of “threaten” rather than “-complicate” as it is a more accurate description of what the Eagle Drone is doing.

Table H.27 analyses the HazTOPS scenario where agitated local people use lasers to disrupt the Eagle Drone's security patrol. The **AIC interaction** highlights the drone’s perceived threat to the locals, exacerbating tensions and resulting in **"More" conflict** as the guideword. The scenario describes how residents' dissatisfaction escalates into disruptive laser use, impairing the drone's sensors.

Table H.28 HazTOPS Analysis of agitated_people use lasers scenario

Title	agitated_people use lasers
AIC interaction	**I4:** |{flying_eagle_drone}_[-threaten]_{agitated_people}|
HazTOPS Aspect	**Definition**
Hazards, Threats or Opportunities Scenario: Guide word	More: More conflict with local people
Operating Scenario Context	The security drone performs a restricted security patrol of the train track zone. The agent's presence further agitates the local people.
Hazard, Threat or Opportunity definition (consider used-systems)	Additional conflict due to resident dissatisfaction. They use lasers to interfere with the Eagle Drone patrolling operation.
Foreseeable Sequence of Events	Local people point lasers; Eagle Drones' sensors get disrupted, reducing the effectiveness of monitoring.
Potential harm or benefit	Damage to Eagle Drone. Failure in adversarial drone detection, risk to public safety.

H.6.3.6 Soft Safety Hazard example

While capturing the list of interactions, we re-considered some of the actions in the AIC model; for example, the model shows the police performing a “remove” control action with a neutral effect on the trees. That is because we thought the police may not have a direct intention to do so. This interaction may lead to social uproar and thus indirect impact on the safe deployment of the system. This is an example of a Soft Hazard, and we could identify it immediately. Ethical considerations can be demonstrated objectively as a design aspect in a safety case. The figure below shows the complexity field related to the emergent soft hazard. While they don’t directly impact the safety of the Eagle Drone, they affect the harmony of achieving the overall ideal whole.

.

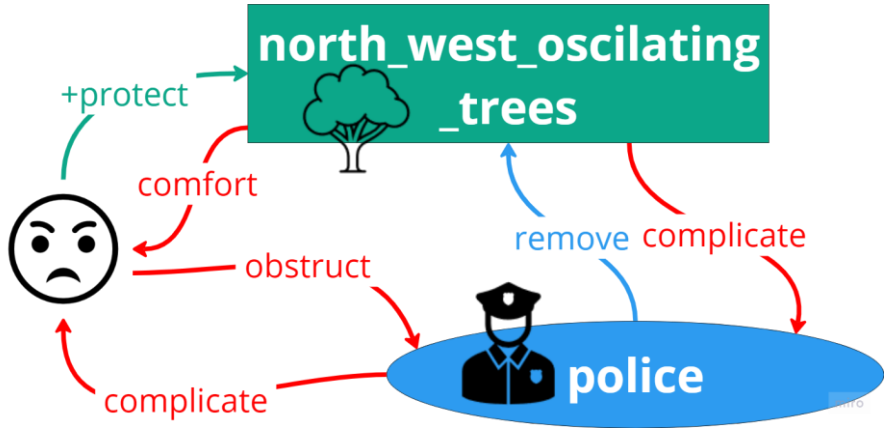


Figure H.19 Soft Hazard Complexity Field Model

H.6.4 Predictive Thinking Pipeline 4: Elicitate AIC System-Level Requirements and Training Requirements

In this step, we will model every derived HazTOPS and define mitigating system-level requirements. We will perform the process in the following 2 examples:

H.6.4.1 Example 1) Mitigating “agitated_people use lasers” HazTOPS:

This example related to the police force problem solution field. Part of it is the Eagle Drone performing restricted patrolling missions.

(a) Step 1) Model the Complex of Interest Operating Scenario Context

At a high level, we model the Eagle Robot Agent as a source, and the AIC relationships with the required used systems to achieve the emergent capability. First, we identify the unresolved influence relationship problem between the agent and Track Zone (TZ) in Figure H.21:

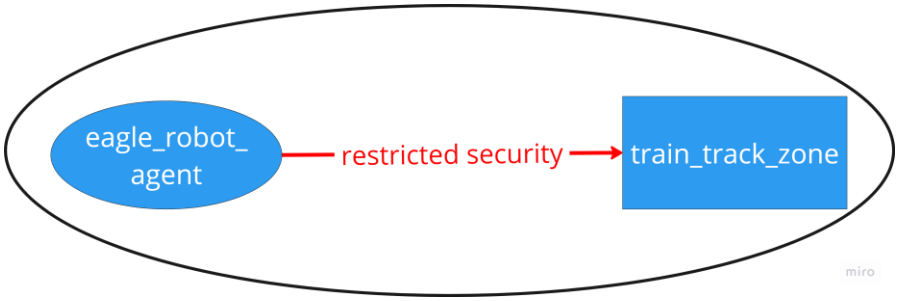


Figure H.20 Initial abstraction of Eagle Robot (ER) Agent emergent capability boundary

We will then need to resolve the influence relationship problem to its Appreciation and Control relationships, rendering the following refined definition of the influential capability in Figure H.21:

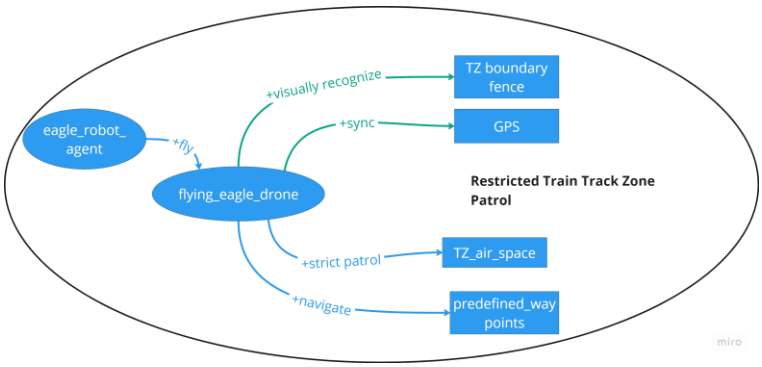


Figure H.21 Restricted TZ security boundary model

Figure H.22 represents a model of the relationships between the Agent and used systems, the composition of which results in the emergence of “Restricted TZ Security” capability. The Agent aims to provide a supportive influence on TZ's purpose by securing TZ within the limit of its boundary lines. To achieve this, it must intelligently control the navigation through waypoints and patrol TZ air space. To successfully navigate through waypoints, it must appreciate GPS by synchronisation and boundary fence through visual recognition to maintain strictness within the TZ boundary fence. We then describe the discovered relationships. Table H.28 details how the AIC actions are associated with AIC relationships in Figure H.22 and the supportive actions from support systems.

Table H.29 AIC-structured interactions detailing the flying Eagle Drone's positioning and patrolling behaviours supported by police monitoring systems

Agent Output Behaviour	**Agent Input Behaviour that delivers Output Behaviour**	**Supportive systems AIC Behaviour**
I1: {eagle_robot_agent}_[+restri	**A1:** {flying_eagle_drone}_[+syncs]_{GPS}	**A3:** {monitoring_police}_[+ dictates]_{predefined_e

cted_security]_{train_track_ zone}	**A2:** {flying_eagle_drone}_[+visually_recogni ze]_{track_zone_ boundary_fence}	agle_drone_patrol_area _ strategy}
	C1: {flying_eagle_drone}_[+ navigates]_{predefined_waypoints} **C2:** {flying_eagle_drone}_[+ patrols]_{TZ_air_space}	

We refine the capability further by considering the agent as a sink and identifying the affecting systems in its surrounding environment **(step 2)**.

(b) Step 2) Model hazards mitigation Ordered-AIC complexity field

From the above HazTOPS Operating Scenario, we discovered that Local People's basic influence purpose (for any relationship they make with any other system in the environment) of preserving comfort in the emergent scenario whereby there is a flying robot within their sphere of concern, may lead to the emergence of acquiring the goal of *obstructive control over Eagle Robot TZ patrol* which may influence them to develop a new capability of distracting Eagle Drone camera functionality using lasers. We abstract the capability into a single agent that is impacting the agitated_people in Figure H.23:

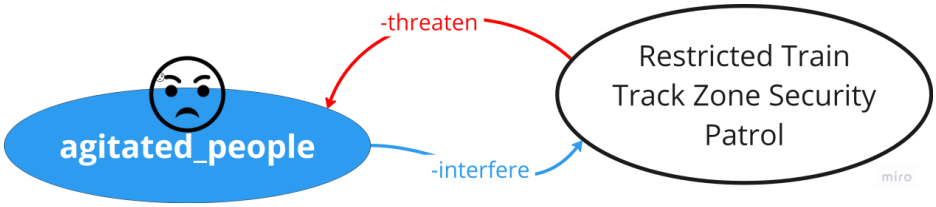


Figure H.22 Abstract model of the agitated people hazard

We need to understand the aspects related to the external agent (agitated_people). To model the external agent complexity field, we can re-interpret (tailor) the general AIC thought process in section 4.4.11 for this situation:

- What complex do the agitated people have no influence or control and influence them to the point of being threatened by it? Answer: Patrol altitude.
- What aspect of agitated people can influence and control while the appreciated aspect also impacts them? Answer: people's threat perception.
- What possible complex that can be easily obtained and controlled mitigates the appreciated aspect? Answer: they can quickly get laser.

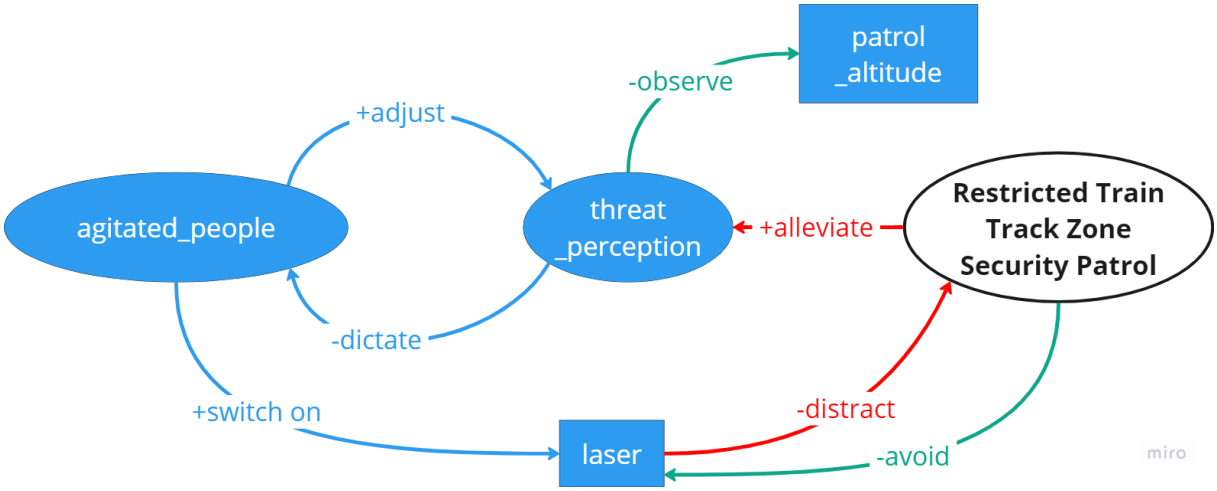


Figure H.23 Refined capability interactions with local people (LP) threat perception

In Figure H.24, we introduced LP threat perception as a cognitive agent within LP agent control, which has a feedback effect on LP behaviour. We also introduced a desired relationship we want to design between our agent and Local People, which has a supportive influence goal to comfort their worries. Similarly, the laser has an obstructive influence over the Eagle Drone agent. In summary, we have two influence relationships, which must be resolved into corresponding Appreciation and Control relationships.

To refine the model better, we need to abstract unnecessary detail. In Figure H.25, we abstracted LP agent and threat perception into a more holistic agent called “LP threat perception” to focus on what we should be concerned about in this relationship.

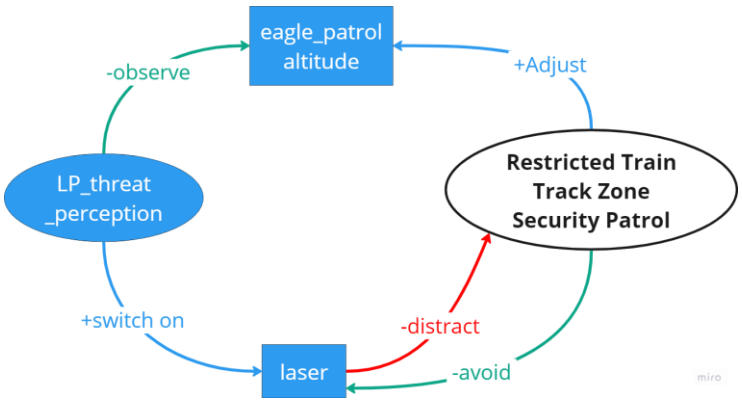


Figure H.24 Further refined relationship between Agent Capability and Local People(LP)

We are primarily concerned about LP threat perception and not LP themselves and reducing the complexity of the model effectively. We may not abstract it if we consider doing something about LP themselves. Also, we reduced model complexity. We also resolved the “comfort” influence relationship with two counter relationships to Local People influence purpose. One is a supportive control relationship with Eagle Patrol altitude, and the other is an obstructive appreciative relationship to Laser attack. In other words, comfort should be achieved by adjusting the altitude of the patrol to a level where the Local People find it comfortable, and lasers should be avoided by some capabilities that allow for the detection of laser attacks.

However, we still don't know how to design this appreciative relationship (between laser and Agent). In section 6.6, we define the intrinsic hard hazard of appreciation. Here, we note that our agent is in an appreciative situation with respect to the laser, which makes it vulnerable. We notice an unresolved relationship between laser and the agent capability defined as "distract". We need to refine this relationship further to understand how to achieve the obstruction of laser impact of distracting our Agent. We do so in the subsequent refinement presented in Figure H.26. In Figure H.26, we resolved all influence relationships into their corresponding Appreciation and Control relationships, thus harmonising the agent relationship with the Local People. This marked the relative end of the refining process, as the relationship can be considered harmonious.

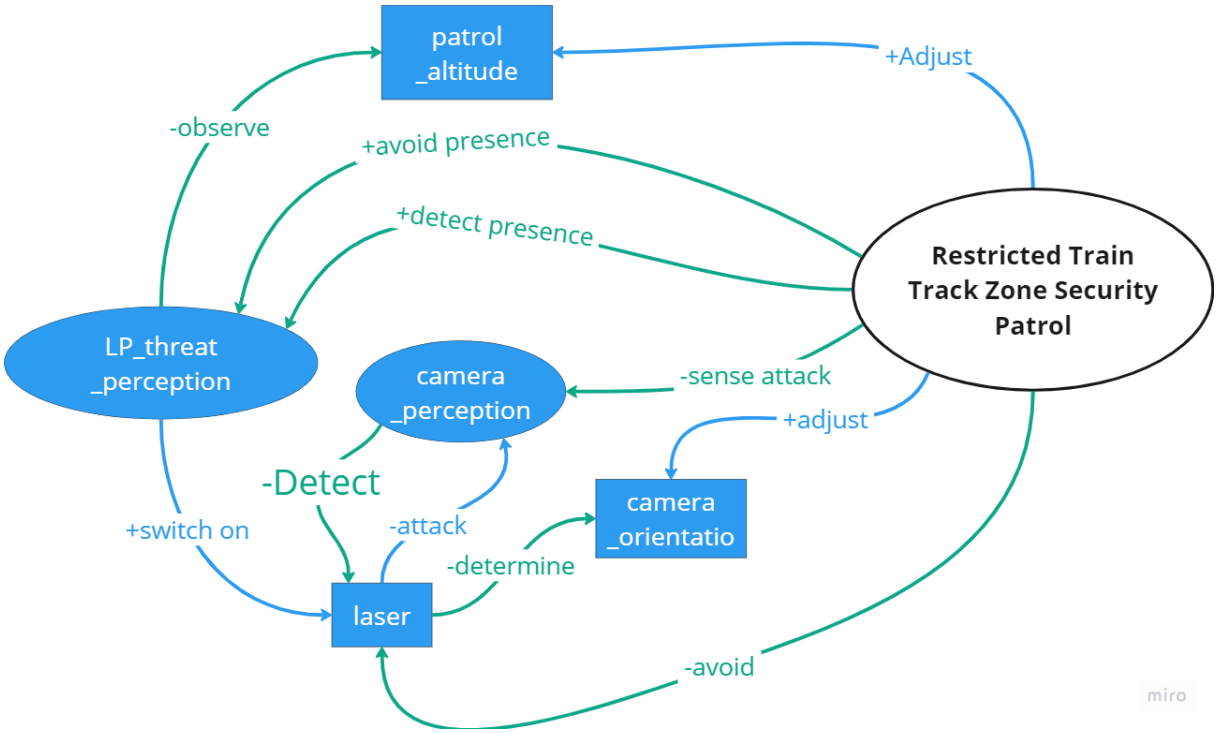


Figure H.25 Final refinement of the Local People and the Agent relationship resolves the restricted security emergent capability.

Finally, we map the newly discovered action descriptions to the original HazTOPS Operating Scenario in Table H.30.

Table H.30 HazTOPS Resolution Matrix

HazTOPS Title	agitated_people use lasers	
Agent target I-interactions	**Agent AC-interactions**	**Actions from supportive systems**
Target Influence: **I1:** {flying_eagle_drone}_[+strict_	**Appreciation:** **A1:** {flying_eagle_drone}_[+syncs]_{GPS}	**A3:** {monitoring_police}_[+dictates]_{predefined_eagle_drone_patrol_area_ strategy}

patrol]_{track_zone_ boundary}	**A2:** {flying_eagle_drone}_[+visually_recognize]_{track_zone_ boundary_fence} **A4:** {flying_eagle_drone}_[-senses]_{laser_attack} **A5:** {flying_eagle_drone}_[-avoid]_{laser_attack} **A6:** {moving_camera_percpetion}_[-detect]_{laser_attack}	**A7:** {monitoring_police}_[-porhibit]_{friendly_laser_attacks} **A8:** {monitoring_train_network}_[-porhibit]_{friendly_laser_attacks}
	Control: **C1:** {flying_eagle_drone}_[+navigates]_{predefined_waypoints} **C2:** {flying_eagle_drone}_[+adjust]_{camera_orientation} **C3:** {flying_eagle_drone}_[+patrols]_{TZ_air_space}	
I2: {flying_eagle_drone}_[+alleviates]_{LP_threat_perception}	**A9:** {flying_eagle_drone}_[+detect presence]_{local_people} **A10:** {flying_eagle_drone}_[+avoid presence]_{LP_threat_perception}	
	C4: {flying_eagle_drone}_[+adjust]_{patrol_altitude}	

Table H.30 outlines a structured framework for resolving the HazTOPS scenario titled "agitated_people use lasers," focusing on the interactions between the flying Eagle Drone, the operating environment, and supportive systems. The table categorises behaviours into Target

Influence (I), Appreciation (A), and Control (C) interactions, demonstrating how the system mitigates the risks associated with laser attacks by agitated individuals.

- Target Influence (I):
    - I1: The flying Eagle Drone conducts strict patrols along the track zone boundary (|{flying_eagle_drone}_[+strict_patrol]_{track_zone_boundary}|) to monitor and secure the area.
    - I2: The drone helps reduce the threat perception among local people (|{flying_eagle_drone}_[+alleviates]_{LP_threat_perception}|) by addressing their concerns.
- Appreciation (A):
    - The Eagle Drone relies on GPS synchronization (A1) and visual recognition of track zone boundaries (A2) to maintain its patrol strategy.
    - It senses and avoids laser attacks (A4, A5) and detects laser interference in its camera system (A6).
    - Supportive systems, such as monitoring police and the train network, help prohibit friendly laser activities (A7, A8) to prevent confusion or accidental disruptions.
    - The Eagle Drone also detects and avoids interactions with local people (A9, A10) to reduce tension.
- Control (C):
    - The drone navigates predefined waypoints (C1) and adjusts its camera orientation (C2) to counter laser interference.
    - It patrols the airspace above the train zone (C3) and adapts its patrol altitude (C4) to mitigate threats effectively.

This table demonstrates a comprehensive resolution strategy that integrates autonomous system behaviours (Eagle Drone actions), environmental factors (laser interference), and support systems (police and train network) to maintain safety and reduce conflict. It highlights the interplay between system autonomy, situational awareness, and operational control to address a complex safety scenario.

(c) Step 3) Ordered-AIC-based Mitigating System or Safety Requirements Derivation (Safe Operating Concept)

In this step, we convert AC actions into systems or safety requirements. So far, we have focused on defining Agent relationships in the given hazardous scenario. We also defined Eagle Drone as the system used by the agent to delegate its actions. In this step, we collate all discovered actions

in a table and convert those actions into Eagle Drone systems requirements considering the within-immediate-reach or within-immediate-reach methods that shall deliver the agent's will. The relevant system requirements are defined from Control and Appreciation actions, with the influence action included at the end of each requirement after the phrase "in order to". Use the following format:

[**Given:** A or C actions] **in order to** [I action], **Then** [mitigation requirement] **In order to** [I action]

For example:

Eagle Drone Safety Requirement:

Safety Requirement 1:

- **Given: C2:** The Eagle Drone adjusts its camera orientation away once the laser is sensed,
- **In order to:** secure the TZ within the TZ boundary (I1),
- **Then:** The Eagle Drone camera system shall turn away from the direction where the laser was detected and then return to the original position.
- **In order to:** secure the TZ within the TZ boundary (I1).

(d) Step 4) Extended Concrete Safety, Systems requirements and ML Safety-Training Concept

This step will attempt to derive any required training strategies or sub-system-level concrete requirements. Given **Safety Requirement 1** as an example, we will take a system interaction and attempt to define the supporting components and functionalities that make it happen. For example:

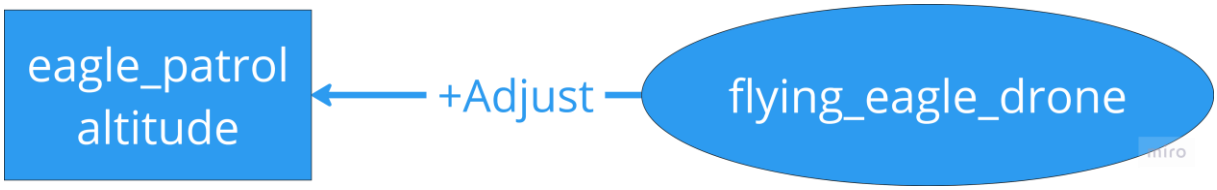


Figure H.26 **C4:** {flying_eagle_drone}_[+adjust]_{patrol_altitude}

To do so, we will utilise the 4-Hows-&-What (4WnH) process in section 7.8. The following is the output of using the technique. We start with modelling the AIC hierarchical schema to help us imagine what needs to be broken down. We do this process interactively in conjunction with the table. Below is the full AIC schema:

Figure H.28 visually represents the hierarchical breakdown of the flying Eagle Drone's functionality when adjusting its patrol altitude. It is structured around the AIC (Appreciation, Influence, Control) framework and employs the 4WnH technique. The diagram systematically explores how the drone adjusts its patrol functionality to achieve stable operations. Each layer of the schema defines the sequence of interactions between subsystems, highlighting control

actions (blue), appreciation tasks (green), and the physical or behavioural outcomes they aim to influence (red).

- **Influence:** The drone influences its environment by adjusting its altitude during patrols. This begins with **low patrolling altitude** adjustments and progresses through key systems, such as the adaptive flight controller, rotor speed, and energy distribution.
- **Control:** Control actions regulate key processes, such as the flight controller for altitude adjustments, the rotor speed for lift generation, and power modulation to balance energy across all motors.
- **Appreciation:** Sensors and monitoring systems (e.g., GPS, barometric sensors, and motor temperature sensors) enable the drone to appreciate environmental and system situations, ensuring stability and effectiveness in patrol operations.

The schema provides an interactive and layered understanding of the system's behaviour, highlighting interdependencies between influence, control, and appreciation.

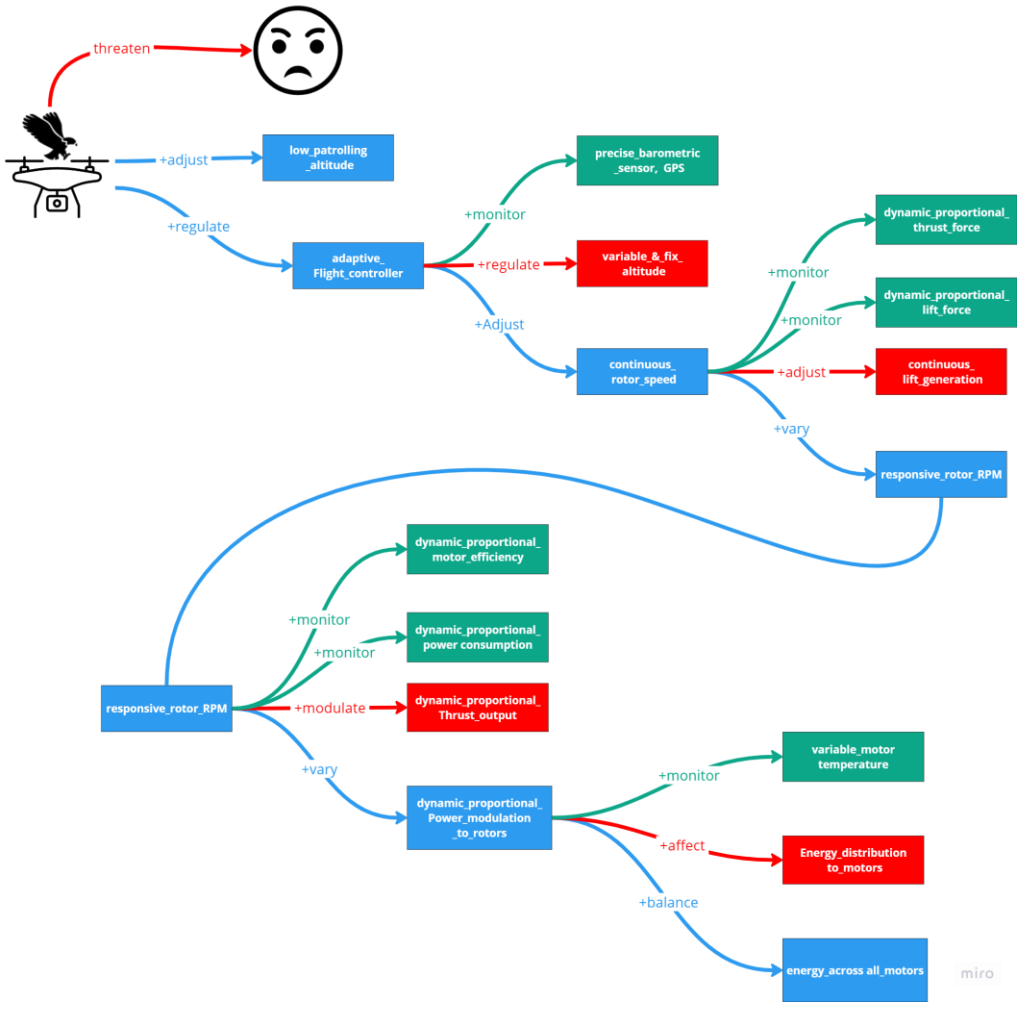


Figure H.27 AIC hierarchical modelling schema for 4WnH analysis

Table H.30 complements Figure H.28 by detailing the 5-HnWs analysis for the Eagle Drone's patrol functionality. It defines source situations, their interactions with target subsystems (sink situations), and how influence is controlled and appreciated:

1. Source Situation: The starting point is the flying Eagle Drone adjusting its patrol altitude, influenced by the adaptive flight controller.
2. How to Control It (Control): Control actions are listed for each step, such as regulating the flight controller, adjusting rotor speed, and modulating power across motors to ensure effective patrol adjustments.
3. What to Appreciate (Appreciation): Appreciation involves monitoring critical factors like GPS data, barometric sensors, thrust force feedback, motor efficiency, and motor temperatures to maintain operational effectiveness.
4. Interactions: Each layer of control and appreciation directly influences the sink situations, ensuring stability and responsiveness in altitude and energy management.

Table H.31 4WnH process for Eagle Drone adjusting patrol functionality

What Source situation?	**What interaction?**	**Sink situation (What to influence)**	**How to control it? (control)**	**What to appreciate? (appreciation)**
Flying_eagle_drone	+adjust	low_patrolling _altitude	How 1? +Regulate adaptive_ Flight_controller	What 1? +Visually scanning visible_terrain, GPS
Adaptive_Flight _controller	+Regulate	Variable_&_fix_ altitude	How 2? +Adjust continuous_ rotor_speed	What 2? +Monitor precise_barometric _sensor, GPS
continuous_ rotor_speed	+adjust	continuous_ lift_generation	How 3? +Vary responsive_ rotor_RPM	What 3? +Measure dynamic_ proportional_thrust, dynamic_proprtional _lift force feedback sensors.
responsive_	+adjust	dynamic_	How 4?	What 4?

rotor_RPM		proportional_ Thrust_output	+Modulate dynamic_ proportional_ Power_modulation _to_rotors	+Monitor dynamic_ proportional_ power consumption and dynamic_ proportional_ motor efficiency.
dynamic_ proportional_ Power_modulation _to_rotors	+affect	Energy_distribution to_motors	How 5? +Balance energy_across all_motors	What 5? +Monitor variable_motor temperature

Step 4.1) ML Safety-Training Requirement derivation (Training Concept):

Part of satisfying Safety Requirement 1 is training the Eagle Drone to recognise laser attacks. Thus, we need to specify a training requirement for the ML model. In this case, this is a Safety-Training requirement since it mitigates a safety hazard.

> **ML Safety-Training Requirement 1:** The Eagle Drone's ML component shall be trained to recognise laser attacks and perform avoidance manoeuvres.

Key Insights:

The combined use of the AIC schema and 4WnH process systematically breaks down the drone's patrol adjustment function into actionable steps. This approach clarifies how each subsystem contributes to the overall objective, ensuring precise control over patrol operations and adaptability to environmental conditions. The analysis effectively integrates autonomous and system-level thinking, highlighting interdependencies and critical interactions.

Step 4.2) ML dataset requirements derivation

A dataset training requirement can be derived from the training concept. We use the following structure to define requirements over actual datasets:

Dataset requirement structure:

> The AS ML component [Training/Testing/Black Swan Validation] Dataset shall provide the trainee model with a valuable minimum variety of …

In this case:

> **ML development datasets requirement 1:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of laser-targeted attacks on the drone camera from various angles and colours.

H.6.4.2 Example 2) Adversarial drone behind the fence

Let's take the following HazTOPS scenario:

Table H.32 Adversarial Drone hiding behind fence HazTOPS

Title	Adversarial drone behind fence
AIC interaction	**A3:** |{flying_eagle_drone}_[-recognise]_{ visible_adv_drone_visual_appearance}|
HazTOPS Aspect	**Definition**
Hazards, Threats or Opportunities Scenario: Guide word	Less: Adversarial drones are less visible
Operating Scenario Context	Operational environment difficulty grade A. Eagle Drone conducting a usual patrol. The area adjacent to the train track zone fence is not part of Eagle Drone training data.
Hazard, Threat or Opportunity definition (consider used-systems)	Adversarial drones hide behind train track fence
Foreseeable Sequence of Events	The adversarial drone cunningly lands on the floor behind the fence; the Eagle Drone does not notice its presence. A train passes, and the adversarial drone launches to strike the train; the Eagle Drone fails to act swiftly enough.
Potential harm or benefit	Train derailment
Risk Impact	**Likelihood:** Rarely Likely (1) **Concern:** Rarely Concerning (1) **Risk Impact:** Critical (19)
Surprise (1/risk impact)	100% Shocking (black swan event)

(a) Step 1) Model the Complex of Interest Operating Scenario Context

We will start with an abstract relationship between adversarial drones and the Eagle Drone patrolling capability. Initially, we considered the “detect” action as an appreciative interaction.

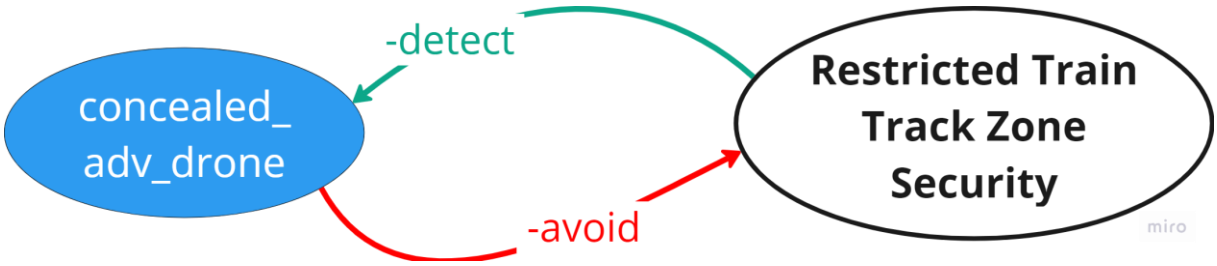


Figure H.28 Problem context definition for concealed adversarial drone

(b) Step 2) Model hazards mitigation ordered-AIC complexity field

Refine the model by introducing the hazard and mitigating solutions, then update the interactions table. We refine it further by resolving influence interaction into AC components, thus replacing the influence interaction "-avoid".

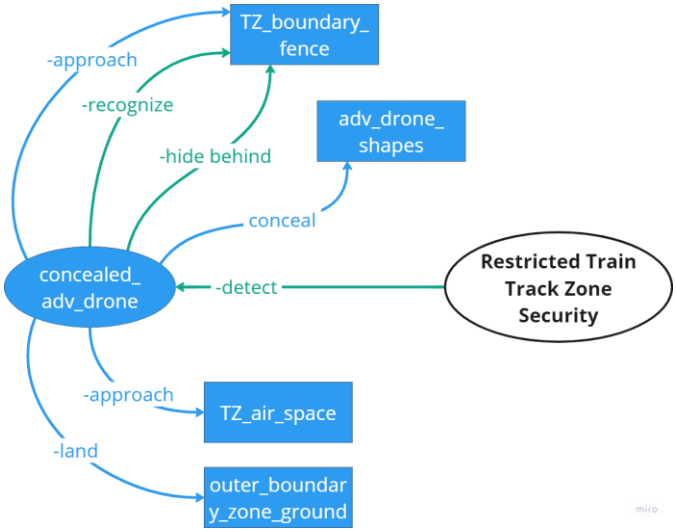


Figure H.29 Refining the concealed adversarial drone problem context

Then, we refine it further by defining Eagle Drone patrolling capability to mitigate actions.

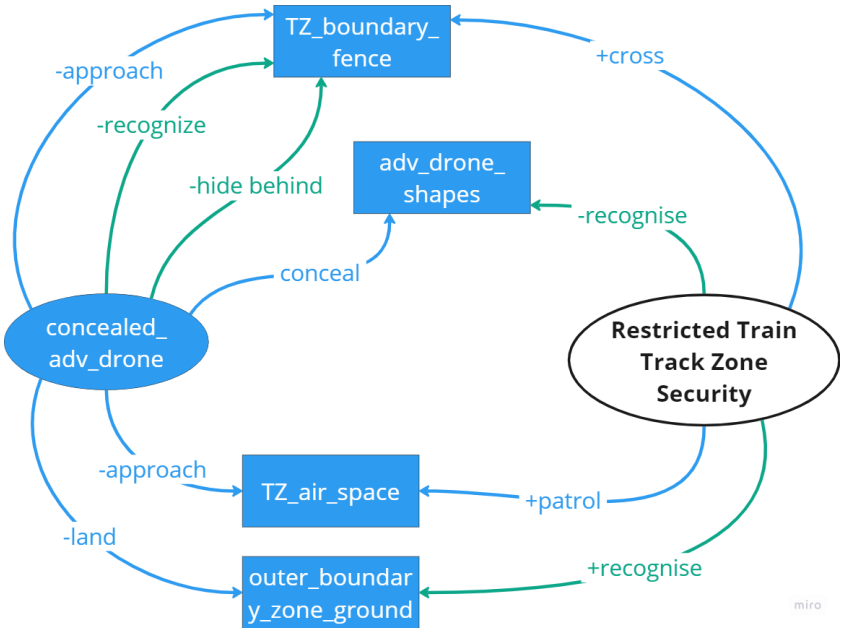
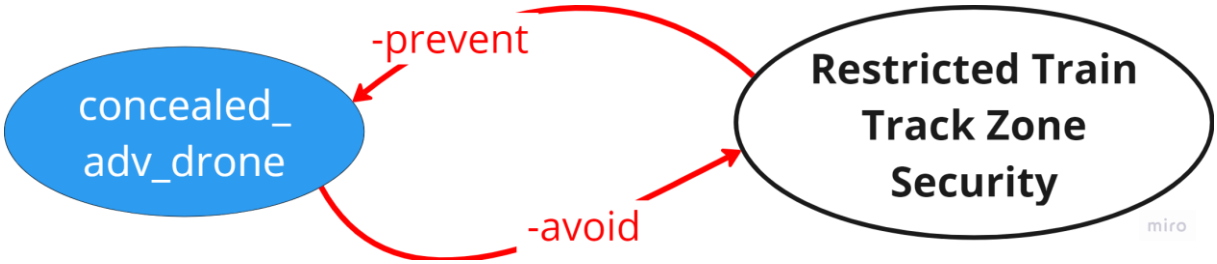


Figure H.30 Mitigating the Concealed Drone problem scenario

However, as we progressed to define the AIC interactions, we realised that the "-detect" interaction between the Eagle Drone and adversarial drone should not be an appreciative interaction but rather an influence interaction of "-prevent". This is particularly true since the detection of adv_drone influences its behaviour and decision-making process. So we updated the initial interaction into an influence-influence design pattern:



Here is where we would update the AIC interaction table:

Agent Output Behaviour	**Agent Input Behaviour that delivers Output Behaviour**	**Supportive systems AIC Behaviour**
I1: {flying_eagle_drone}_[+strict_patrol]_{track_zone_ boundary}	**A1:** {flying_eagle_drone}_[+syncs]_{GPS} **A2:** {flying_eagle_drone}_[+visually_recognize]_{track_zone_ boundary_fence}	**A3:** {monitoring_police}_[+dictates]_{predefined_eagle_drone_patrol_area_ strategy}
	C1: {flying_eagle_drone}_[+navigates]_{predefined_waypoints} **C2:** {flying_eagle_drone}_[+patrols]_{TZ_air_space}	
I3: {flying_eagle_drone}_[-prevent]_{adversarial_drone}	**A11:** {flying_eagle_drone}_[+recognise]_{adv_drone_shapes} **A12:** {flying_eagle_drone}_[+recognise]_{boundary_zone_ground}	**A13:** {monitoring_police}_[+dictates]_{predefined_eagle_drone_incrusion_outside_boundary_fence}
	C2: {flying_eagle_drone}_[+patrols]_{TZ_air_space} **C5:** {flying_eagle_drone}_[+cross]_{track_zone_boundary_fence}	
I4: {concealed_adv_drone}_[-avoid]_{flying_eagle_drone}	**A13:** {concealed_adv_drone}}_[-recognises]_{track_zone_ boundary_fence}	

	A14: {concealed_adv_drone}_[-hide behind]_{track_zone_ boundary_fence}	
	C6: {concealed_adv_drone}_[+conceal]_{ adv_drone_shapes} **C7:** {concealed_adv_drone}_[-approach]_{track_zone_ boundary_fence} **C8:** {concealed_adv_drone}_[-approach]_{track_zone_air_space} **C9:** {concealed_adv_drone}_[-land]_{outer_boundary_zone_ground}	

(c) Step 3) Ordered-AIC-based System or Safety Requirements Derivation (Safe Operating Concept)

In this step, we convert the identified actions into detailed system requirements. We have comprehensively articulated the problem domain through extensive analysis and discussion by this point in the process. We have objectively modelled the problem domain as we think it would be, identified key user needs, and thoroughly examined existing constraints. Our requirements are now firmly based on the well-defined parameters of the problem, ensuring that every aspect is aligned with a real-world aspect and goals and can be effectively translated into actionable specifications for system development. Below is an example of the requirements:

Table H.33 Safety requirements derivations to mitigate the concealed drone problem.

AC interaction	**Mitigating Safety or Systems Requirements (Safe Operating Concept)**
A11: {flying_eagle_drone}_[+recognise]_ {adv_drone_shapes}	**Safety Requirement 2: Eagle Drone recognises approaching adversarial drone shapes.** **Given:** adversarial drone recognises track zone boundary fence, adversarial drone approaches track zone airspace, **In order to:** avoid Eagle Drone detection and intrusion prevention, **Then:** the Eagle Drone shall recognise approaching adversarial drone shapes,

	In order to: prevent adversarial drone incursion.
A12: {flying_eagle_drone}_[+recognise]_ {outer_boundary_zone_ground}	**Safety Requirement 3: Eagle Drone recognises adversarial drones during approach and landing** **Given:** adversarial drone landed on outer boundary zone ground, **In order to:** avoid Eagle Drone detection and intrusion prevention, **Then:** 3.1: The Eagle Drone shall recognise adversarial drone approach to the fence during landing. 3.2: The Eagle Drone shall recognise an adversarial drone landed on the outer ground zone. **In order to:** prevent adversarial drone incursion.
C5: {flying_eagle_drone}_[+cross]_ { track_zone_boundary_fence} **A13:** {monitoring_police}_[+ dictates]_{predefined_eagle_drone_ incrusion_outside_boundary_fence}	**Safety Requirement 4: Police offer sets predefined patrolling coordinates** **Given:** adversarial drone recognises track zone boundary fence, adversarial drone approaches track zone airspace, adversarial drone landing outside track zone fence, adversarial drone landed outside track zone fence, adversarial drone attempts to conceal shape, **In order to:** avoid Eagle Drone detection and intrusion prevention, **Then:** 4.1: The monitoring police officer shall set predefined distances across the fence for extended patrolling. 4.2: The Eagle Drone shall follow a predefined cross-boundary distance during patrol to monitor for any landing or about to land adversarial drones, **In order to:** prevent adversarial drone incursion.

Table H.32 outlines the safety requirements derived to mitigate the concealed adversarial drone problem through specific Appreciation (A) and Control (C) interactions. The first requirement (Safety Req 2) mandates that the Eagle Drone recognise the shapes of adversarial drones approaching the track zone airspace to prevent intrusion. Safety Req 3 further specifies that the

Eagle Drone must detect adversarial drones while landing in the outer boundary zone, ensuring comprehensive coverage of potential entry points. Lastly, Safety Req 4 focuses on extending the Eagle Drone's patrol capabilities beyond the track zone boundary fence, as dictated by the monitoring police controller, to detect adversarial drones attempting to conceal themselves or land just outside the boundary. These requirements collectively emphasise proactive recognition and extended surveillance to prevent adversarial drone incursions effectively.

(d) Step 4) Extended Concrete Safety, Systems requirements and ML Safety-Training Concept

In this example, we will take the following safety requirements:

Safety Requirements 3: Eagle Drone recognises adversarial drones during approach and landing.

Given: adversarial drone landed on outer boundary zone ground,

In order to: avoid Eagle Drone detection and intrusion prevention,

Then:

3.1: The Eagle Drone shall recognise adversarial drone approach to the fence during landing.

3.2: The Eagle Drone shall recognise an adversarial drone landed on the outer ground zone.

In order to: prevent adversarial drone incursion.

Note that requirements 3.1, 3.2 are related to the recognition of an adversarial drone. This is an ML training issue rather than a mechatronic issue. The architect is free to use 4WnH process to derive the ML training safety requirement. However, we will refrain from doing so in this example for variety.

Step 4.1) ML Safety-Training Requirement derivation (Training Concept):

Part of satisfying Safety Requirement 3 is training Eagle Drone to recognise adversarial drones approaching and landing on the ground. Thus, we need to specify a general training requirement for the ML model, which we refer to as the Training Concept. In this case, this is a Safety-Training requirement since it mitigates a safety hazard. To do so, we used the following structure:

ML Safety-Training Requirement n: [system of interest] ML component shall be trained to [training experience].

In this case:

Table H.34A ML Safety Requirements Derivation

Mitigating Safety or Systems Requirements (Safe Operating Concept)	**ML Safety-Training Requirements (Training Concept)**
3.1: The Eagle Drone shall recognise the adversarial drone approach to the zone boundary and the fence.	**ML Safety-Training Requirement 2:** The Eagle Drone's ML component shall be trained to recognise adversarial drones approaching the train tracks zone boundary, fence, and train and attempting to land.
3.2: The Eagle Drone shall recognise an adversarial drone landed on the outer ground zone.	**ML Safety-Training Requirement 3:** The Eagle Drone's ML component shall be trained to recognise an adversarial drone landed on the outer ground zone.

Step 4.2) ML dataset requirements derivation

Thus, a dataset training requirement can be derived from the training concept. We use the following structure to define requirements over actual datasets:

Dataset requirement structure:

> The [system of interest] ML component [Training/Testing/Black Swan Validation] Dataset shall provide the trainee model with a valuable minimum variety of..

In this case:

> **ML development datasets requirement n**: The Eagle Drone ML component [Training/Testing/Black Swan Validation] Dataset shall provide the trainee model with a valuable minimum variety of..

Table H.35B ML ML Safety-Training Dataset Requirement derivation

ML Safety-Training Requirements (Training Concept)	**ML Safety-Training Dataset Requirement**
ML Safety-Training Requirement 2: The Eagle Drone's ML component shall be trained to recognise adversarial drones approaching the train tracks zone boundary, fence, and train and attempting to land.	**ML development datasets requirement 2:**The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones (typical off-the-shelf drones) approaching the train, train tracks fence and attempting to land next to the train tracks

	boundary. **In order to:** train the drone's machine learning algorithms to dynamically identify, track, and trace adversarial drones during typical operations.
ML Safety-Training Requirement 3: The Eagle Drone's ML component shall be trained to recognise an adversarial drone landed on the outer ground zone.	**ML development datasets requirement 3:**The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones (typical off-the-shelf drones) landed next to the train tracks boundary. **In order to:** train the drone's machine learning algorithms to dynamically identify, track, and trace adversarial drones during typical operations.

H.7 Stage 3B: Comprehensive Operational Environment Definition[5]

For this application, we reduced the comprehensive ODD classification into the following categories and considered Grade A uncertainty. For a detailed description of how to define ODD, refer to sections 4.7 and 4.8.

Table H.36 Operational Design Definition for Eagle Robot Deployment in Train Track Zone

System of interest		Eagle Robot
Solution Operational Space		Train Track Zone in the UK
ODD Uncertainty Grade		Grade D
Development purpose		Training and Testing
Environment Characteristics		**Less** favourable natural environment system
Natural Lighting Conditions		Cloudy / Overcast
Weather Conditions	**Precipitation** mm/h	20-50
	Wind km/h	20 - 30

[5] See also section 6.5

	Humidity %	0 – 20 & 70 - 90
	Visibility km	0.4 - 1
	Cloud Cover	Mostly Cloudy or Cloudy (5/8 to 7/8 oktas)
	Snow mm/12 hrs	150 – 250
	Pollen	Very high
	Sand	Sandstorm
	Temperature	To be defined by the architect
	Sunshine Duration	4 hours
Time of the Year: Seasons-specific environmental characteristics		3 types
Landscapes type variety definition		7 types
Geographical region-specific natural phenomena		7 types
Time of the Day		6 types
Perceived Horizon Attitude		5 types
Sun sphere positioning		3
Moon sphere positioning		3
Specialised zones features		3+ features

This table defines the operational design domain (ODD) and environmental characteristics for deploying the Eagle Robot within the train track zone in the UK under **Grade D uncertainty**. We may choose different sets for ODD when developing testing datasets. However, we will keep the same ODD for generating the datasets in this application.

H.8 Stage 4: Disordered AIC-Driven Black Swan Scenarios Prediction[6]

In this stage, you may consider disordered AIC timing (before or after). Consider section 6.4 (AIC Timing). Some of the output of this stage is to generate Black Swan validation datasets for validating ML components to handle Black Swan scenarios. However, nothing may stop the architect from dedicating some of the black swan scenarios to be part of the training and testing process of the ML component.

[6] See also section 6.6

At first, we choose the complexity field we intend to perform the AIC perspective shift. For this we will select the last complexity field in Figure H.31:

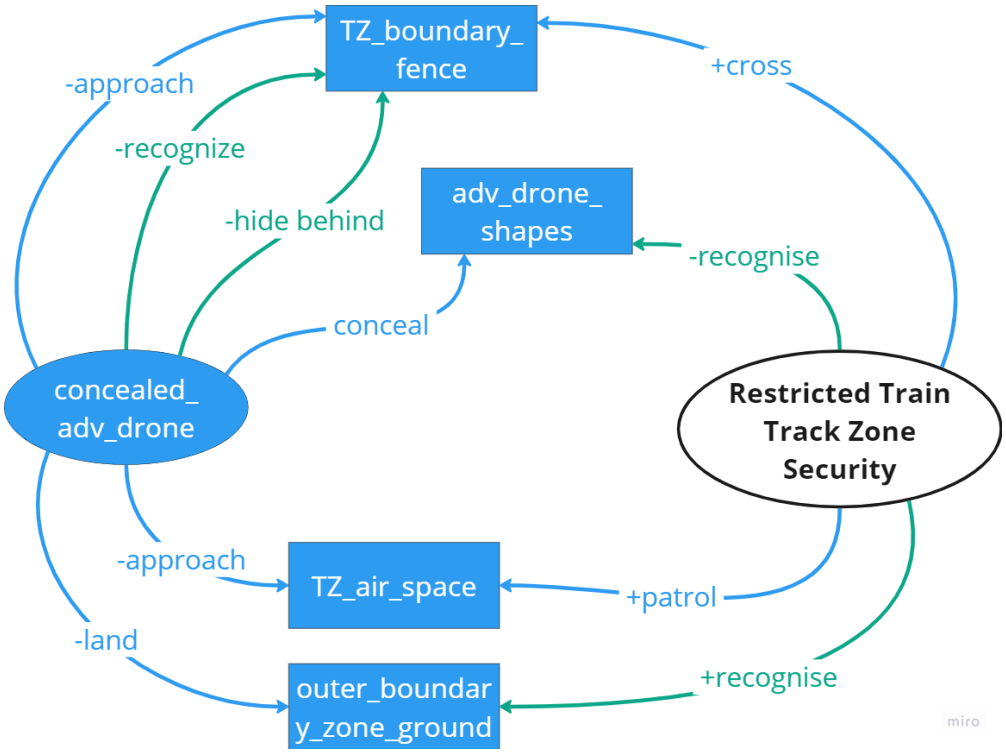


Figure H.31 Example complexity field

To implement the AIC perspective shift, we need to perform the following process:

H.8.1 Step 1) Define the interactions

Define the interactions needed to predict a potential emergence. We chose the following relationship:

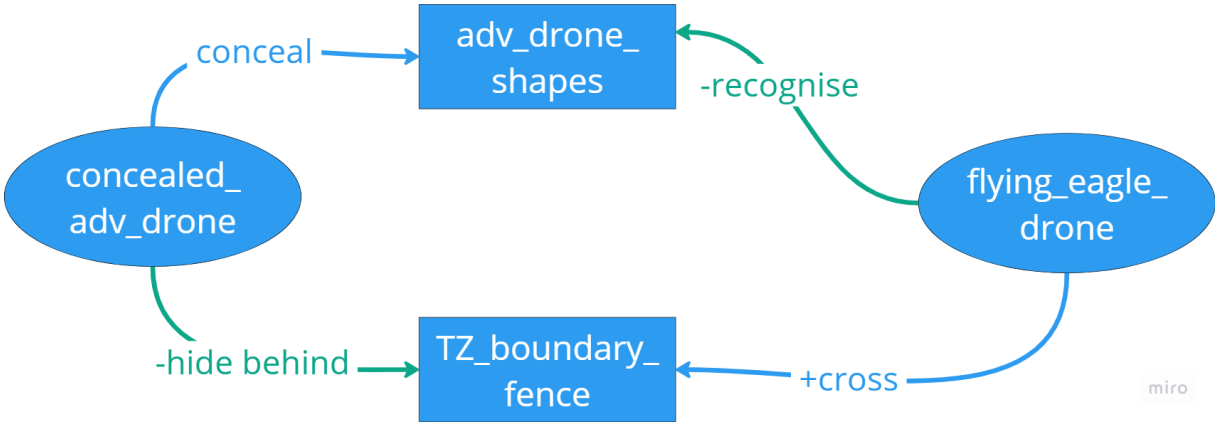


Figure H.32 Example relationship of interest to perform the perspective shift

To increase the detailed context of the relationship, we need to extend interactions with the reactions between each node.

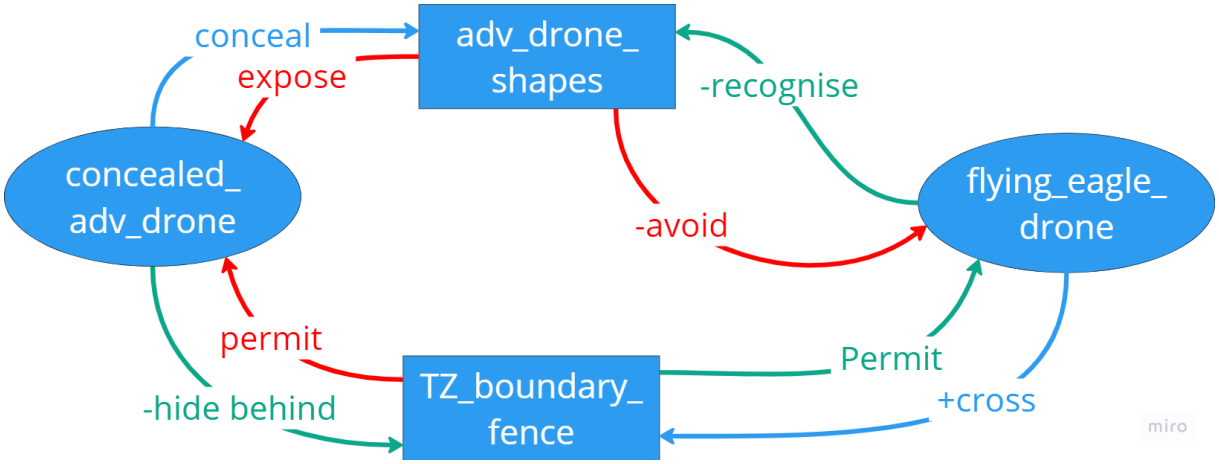


Figure H.33 Extending the interactions with reactions among systems (binary relationships)

H.8.2 Step 2) Define the ArcMatrix

Define the interaction's current AIC factors using the ArcMatrix (section 7.2.1). We choose the binary relationship between the Eagle Drone and adversarial drone shapes.

Table H.37a Deep AIC factorisation for Eagle Drone and adversarial drone shapes

	Flying_eagle_drone	**Adv_drone_shapes**
Flying_eagle_drone		**Supra Source:** Train Network. **PrimeP:** Safely transport people and goods. **Goal:** detect adversarial drone presence. **Goal type:** Appreciation. **Action:** recognise adversarial drone shape. **Action type:** Appreciation. **Effect:** Obstructive.
Adv_drone_shapes	**Supra Source:** Adversarial Scheme. **PrimeP:** Disrupt Train Network operations. **Goal:** avoid detection by Eagle Drone. **Goal type:** influence. **Action:** obscure Eagle Drone perception.	

	Action type: Influence. **Effect:** Obstructive.	

H.8.3 Step 3) Perform the Perspective Shift

Perform the perspective shift using AIC perspective shift SECoT. In this step, we choose an appropriate perspective shift that we believe will be a potential black-swan event not foreseen during modelling the interaction between adv_drone_shapes and flying_eagle_drone. We will select the following shift:

- **AIC type shift.**

For this, we will apply the following Thought Step from SECoT_3:

General Systems Rule: Given a source or sink, it is possible that over time and with the change of complexity, the AIC goals are altered, leading to a new situation of complicatedness.

Predictive question: What would happen if the interaction flow and effect type remained the same, but the goal's AIC type and action type were altered in the future?

Guiding Prompt: Review the AIC dynamics of the observed complexes and alter the nature of the goal and action. Then, define an appropriate action to bear alternative AIC types and describe a scenario in the shifted context.

Completion criteria: The step is considered complete when a scenario demonstrating an alternative AIC goal is detailed.

As per the following schema:

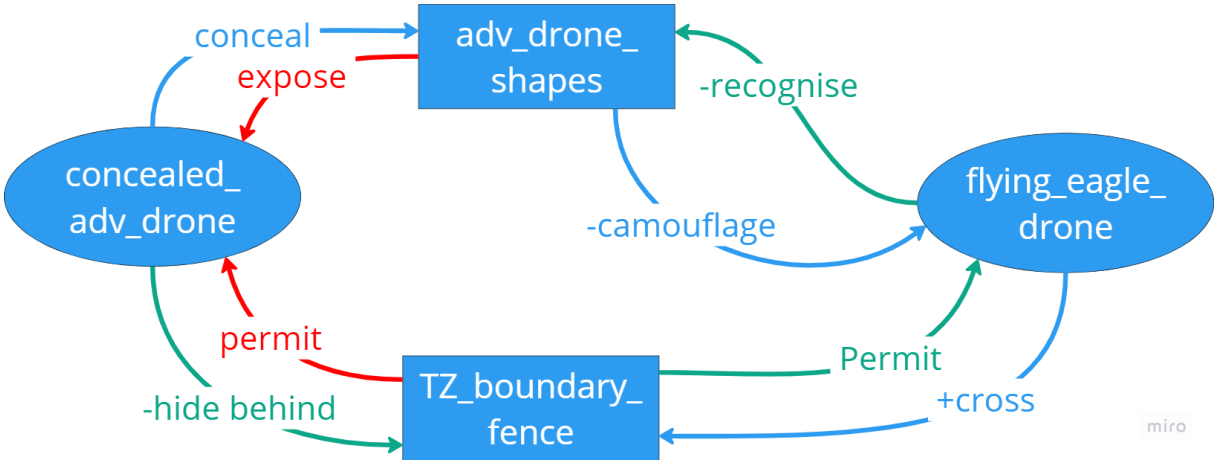


Figure H.34 Shifted perspective from influence to control

The above steps can be captured in the following table:

Table H.36b AIC-type perspective shift

Interaction	**A11:** {flying_eagle_drone}_[+recognise]_{adv_drone_shapes}

AIC factors	Initial perspective (prior)	Shifted Perspective (posterior)
Source	Adv_drone_shapes	Adv_drone_shapes
Sink	Flying_eagle_drone	Flying_eagle_drone
Supra Source	Adversarial Scheme.	Adversarial Scheme.
PrimeP	Disrupt Train Network operations.	Disrupt Train Network operations.
Source's Goal	**obscure Eagle Drone perception.**	**Obscure Eagle Drone perception.**
Source's Goal type	***Influence***	***Control***
Source's Action	***Avoid***	***Camouflage***
Source's Action type	***Influence***	***Control***
Source action effect on sink	Obstructive	Obstructive

How do we implement the SECoT_3 thought step above in this context?

General Complex Rule Application:

Over time, as the adversarial scheme evolves in complexity, the adversarial drone's AIC goals may transition from an influence-driven interaction to a control-oriented strategy. This progression alters the interaction flow and introduces a new situation of complexity, requiring re-evaluation of the system's capabilities and countermeasures.

Predictive Question Answer:

If the **interaction flow** and **effect type** remain obstructive, but the **goal's AIC type** shifts to control, the Eagle Drone's effectiveness will diminish significantly. The adversarial drone's adoption of control-based actions will result in undetected incursions, compromising train network security.

Guiding Prompt Application:

- **Reviewing AIC Dynamics**:

- The AIC framework (Appreciation, Influence, Control) was used to examine the interactions between adversarial and Eagle Drones.
- Initially, the adversarial drone's goal was identified as an influence-type action aimed at avoiding detection by the Eagle Drone, relying on evasion and passive tactics.
- The interaction dynamics were revisited to explore how the source (adversarial drone) might escalate its tactics to achieve a more dominant role in the interaction.

- **Altering the Nature of Goal and Action:**

- The adversarial drone's goal is predicted to evolve from an influence-type goal (obscuring detection) to a control-type goal (neutralising the Eagle Drone's ability to monitor).
- One way for this to happen is if the adversarial drone action shifted from avoidance (a passive influence strategy) to camouflage (an active control strategy), using advanced technology to blend with the environment.

- **Defining Appropriate Actions for Alternative AIC Types**:

- With the goal and action redefined, the scenario introduced alternative AIC types for the adversarial drone. Advanced camouflage technology was identified as a potential method for complete invisibility to the Eagle Drone's sensors.
- This action capitalises on the adversarial drone's capacity to render itself undetectable, effectively neutralising the Eagle Drone's monitoring capabilities.

- **Describing the Shifted Context**:

- The scenario depicted how the adversarial drone, using camouflage to blend into the environment (e.g., sky, trees, or train infrastructure), could adapt to evade traditional detection methods employed by the Eagle Drone.
- This shift in tactics highlights the adversarial drone's evolution from a passive strategy (influence) to an aggressive strategy (control) that undermines the Eagle Drone's operational purpose.

Completion Criteria:

The shifted scenario demonstrates an alternative AIC goal where the adversarial drone fully controls the interaction, obscuring its presence and rendering the Eagle Drone ineffective. The mitigation strategy involves enhancing the Eagle Drone's situational awareness and adaptability to counteract the adversarial drone's control-type actions effectively. This ensures the system can respond dynamically to evolving threats and maintain train network security.

H.8.4 Step 4) Predict Harder-to-foresee emergent scenarios (black swan scenario)

In this step, we elaborate on the scenario further to define the sequence of potential actions.
The architect may want to use 4WnH at this stage to help him think deeper about it. Or they may use the 5-whys analysis.

Table H.36c Harder-to-foresee Emergent Scenario (black swan scenario)

Black Swan Scenario	Rationale for prediction: it is possible that …
Black Swan 1: The adversarial drone may shift tactics, using advanced camouflage technology to blend in with environmental elements such as the sky, trees, or even the train infrastructure.	Adversarial drones may evolve their tactics beyond simple evasive manoeuvres and incorporate advanced camouflage technology to avoid detection. With modern developments in adaptive camouflage, such as optical cloaking and AI-driven pattern adaptation, adversarial drones could blend seamlessly into environmental elements like the sky, trees, or train infrastructure.
Black Swan 2: The Eagle Drone, still tasked with monitoring, struggles to adapt to the adversarial drone's increasingly sophisticated tactics, leading to degraded performance in identifying and tracking the adversarial drone.	As machine intelligence evolves, adversarial drones will likely incorporate increasingly complicated evasive tactics to degrade the eagle drone's monitoring performance. These tactics could include AI-driven flight pattern modifications, multi-agent coordinated evasive manoeuvres, or self-modifying behaviour based on the eagle drone's response patterns.

The AIC analysis reveals a shift in the adversarial drone's goal and action type, resulting in significant complexity and interaction dynamics changes. Initially, the adversarial drone aimed to influence the Eagle Drone's perception by avoiding detection. However, the shifted perspective introduces a control-type goal, where the adversarial drone actively seeks to neutralise the Eagle Drone's monitoring capabilities through advanced camouflage tactics.

H.8.5 Step 5) Define mitigating ML Development and Safety Requirements.

Each safety requirement outlined for Black Swan scenarios necessitates corresponding ML component safety requirements to ensure the Eagle Drone's systems can handle unforeseen threats. Below, we define the ML component safety requirements in alignment with each system safety requirement.

Table H.38 Eagle Drone Safety-Training Requirements for Black Swan Scenarios

Black Swan Scenario	**Safety or systems requirements (Safe Operating Concept)**	**ML Safety-Training Requirement (Training Concept)**
Black Swan 1: The adversarial drone may shift tactics, using advanced camouflage technology to blend in with environmental elements such as the sky, trees, or even the train infrastructure.	**Safety Requirement 5: Eagle Drone track and trace camouflaged adv drones on the ground** **Given**: Adversarial drone uses camouflage tactics on the ground, **In order to**: avoid detection by the Eagle Drone, **Then:** The Eagle Drone shall implement machine learning algorithms that can dynamically identify, track, and trace adversarial camouflage tactics on the ground, **In order to**: prevent adversarial drone incursion.	**ML Safety-Training Requirement 4: Recognise Camouflaged drones on the ground** The Eagle Drone's ML component shall be trained to detect, classify, and track adversarial drones using camouflage tactics on the ground by recognising subtle movement patterns, environmental inconsistencies, and spectral differences using AI-driven computer vision models.
Black Swan 1:	**Safety Requirement 6: Eagle Drone track and trace camouflaged adv drones around trees.** **Given**: Adversarial drone uses camouflage tactics among trees, **In order to**: avoid detection by the Eagle Drone, **Then**: The Eagle Drone shall implement machine learning algorithms that can dynamically identify, track, and trace adversarial camouflage tactics with the trees, **In order to**: prevent adversarial drone incursion.	**ML Safety-Training Requirement 5: Recognise camo drones among trees** The Eagle Drone's ML component shall be trained to analyse and differentiate adversarial drones camouflaged within trees by using depth perception models, multi-frame tracking algorithms, and contrast-enhancing neural networks to distinguish hidden objects in complex environments.

Black Swan 1:	**Safety Requirement 7: Eagle Drone employs imaging polarisation image** **Given**: Adversarial drone uses sophisticated camouflage techniques to blend into the environment, **In order to**: avoid detection by the Eagle Drone, **Then**: The Eagle Drone system shall deploy counter-camouflage detection technologies, such as polarisation imaging or acoustic sensors, **In order to**: detect drones that attempt to blend into their surroundings.	**ML Safety-Training Requirement 6: Polarisation imaging and acoustics training** The Eagle Drone's ML component shall be trained to integrate and interpret data from polarisation imaging, thermal imaging, and acoustic sensors to detect drones blending into their surroundings, using sensor fusion techniques to enhance object differentiation.
Black Swan 2: The Eagle Drone, still tasked with monitoring, struggles to adapt to the adversarial drone's increasingly sophisticated tactics, leading to degraded performance in identifying and tracking the adversarial drone	**Safety Requirement 8: Eagle Drone anticipating adv drone behaviours** **Given**: Adversarial drone exhibits complex, evasive behaviours, **In order to**: obscure its movement and avoid detection, **Then**: The Eagle Drone shall incorporate a predictive behaviour model to anticipate the movement and actions of adversarial drones based on previous behaviour and environmental conditions, **In order to**: maintain continuous monitoring and interception.	**ML Safety-Training Requirement 7: Anticipating Adversarial Drone Behaviours Training** The Eagle Drone's ML component shall be trained using predictive behaviour modelling, leveraging supervised learning and historical adversarial drone flight patterns to anticipate evasive manoeuvres and pre-emptively adjust tracking strategies.

Black Swan 2:	**Safety Requirement 9: Eagle Drone switching to alternative monitoring methods.** **Given**: Adversarial tactics neutralise the Eagle Drone's primary monitoring system, **In order to**: hinder the Eagle Drone's ability to track and monitor, **Then**: The Eagle Drone system shall automatically switch to alternative monitoring methods, such as ground-based sensor support, **In order to**: ensure continuous detection and prevent adversarial drone intrusion.	**ML Safety-Training Requirement 8: Monitoring mode adaptation training** The Eagle Drone's ML component shall be trained to recognise anomalies in tracking performance and autonomously switch to alternative *external* detection methods, such as ground-based sensor networks, ensuring continuous surveillance integrity.
Black Swan 2:	**Safety Requirement 10: Eagle Drone maintains comms with ground control and other systems.** **Given**: Adversarial drones are actively attempting to avoid detection by a single drone, **In order to**: disrupt coordinated monitoring, **Then**: The Eagle Drone shall maintain continuous, real-time data exchange with ground-based security systems and other flying drones in the vicinity, **In order to**: provide backup in detecting adversarial drones and enhance monitoring coverage.	**ML Safety-Training Requirement 9: Systemic Awareness Training** The Eagle Drone's ML component shall be trained to process and integrate real-time data from multiple sources, including ground control, cooperative drones, and fixed surveillance systems. AI-driven sensor fusion will be used to maintain coordinated detection and tracking of adversarial drones.

Then for every ML training concept requirement, we derive an appropriate set of ML-component development datasets requirements:

Dataset requirement structure:

The [system of interest] ML component [Training/Testing/Black Swan Validation] shall provide the trainee model with a valuable minimum variety of ..

Table H.39 Black Swans driven dataset requirements

ML Safety-Training Requirements (Training Concept)	**ML development datasets requirements**
ML Safety-Training Requirement 4: Recognise Camouflaged drones on the ground The Eagle Drone's ML component shall be trained to detect, classify, and track adversarial drones using camouflage tactics on the ground by recognising subtle movement patterns, environmental inconsistencies, and spectral differences using AI-driven computer vision models.	**ML development datasets requirement 3:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones using various ground camouflage tactics, such as blending in with, bushes, gravel, soil, trash or pavement next to track zone fence. **In order to**: train the drone's machine learning algorithms to dynamically identify, track, and trace adversarial drones obscured by ground-level camouflage.
ML Safety-Training Requirement 5: Recognise camo drones among trees The Eagle Drone's ML component shall be trained to analyse and differentiate adversarial drones camouflaged within trees by using depth perception models, multi-frame tracking algorithms, and contrast-enhancing neural networks to distinguish hidden objects in complex environments.	**ML development datasets requirement 4:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones using camouflage techniques among various types of trees, including foliage, branches, and tree trunks at different times of day and under varying weather conditions. **In order to**: train the drone to identify, track, and trace adversarial drones that attempt to blend into forested or tree-dense environments.

ML Safety-Training Requirement 6: Polarisation imaging and acoustics training The Eagle Drone's ML component shall be trained to integrate and interpret data from polarisation imaging, thermal imaging, and acoustic sensors to detect drones blending into their surroundings, using sensor fusion techniques to enhance object differentiation.	**ML development datasets requirement 5:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones using advanced camouflage technology, including scenarios where polarisation imaging or acoustic datasets might detect hidden drones. **In order to**: ensure that the drone's counter-camouflage detection algorithms are trained to identify subtle environmental anomalies using polarised lenses and test how advanced adversarial drone camouflage techniques can be detected.
ML Safety-Training Requirement 7: Anticipating Adversarial Drone Behaviours Training The Eagle Drone's ML component shall be trained using predictive behaviour modelling, leveraging reinforcement learning and historical adversarial drone flight patterns to anticipate evasive manoeuvres and pre-emptively adjust tracking strategies.	**ML development datasets requirement 6:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones executing evasive manoeuvres, using different movement patterns, and hiding in complex environments. **In order to**: help the Eagle Drone's predictive behaviour model anticipate the movement and actions of adversarial drones based on their previous behaviour and environmental conditions.

H.9 Stage 5: CuneiForm-based Syllabus for Safety-Driven ML Epistemic Intelligence Development[7]

Outcomes from stages 3 and 4 reveal the epistemic uncertainties faced by the architect and their trainee machine regarding the problem domain. The architect must define the real world and outline potential unexpected scenarios for the trainee machine to meet objectives. Minimising these uncertainties to a reasonable level (ALARP) is crucial, achieved by creating CuneiForms

[7] See also section 6.9

that capture required training regimes. Thus, this process should be termed ML Epistemic Uncertainty Reduction Training, focusing on enhancing the machine's robustness by increasing the variety of real-world scenarios instead of increasing the variety of image augmentations. Dataset augmentation techniques (e.g., colour modifications) will follow, seen as “Aleatoric Uncertainty Reduction” since they alter pixel randomness rather than actual scenarios. The following is a process model that describes the main activities in the strategy:

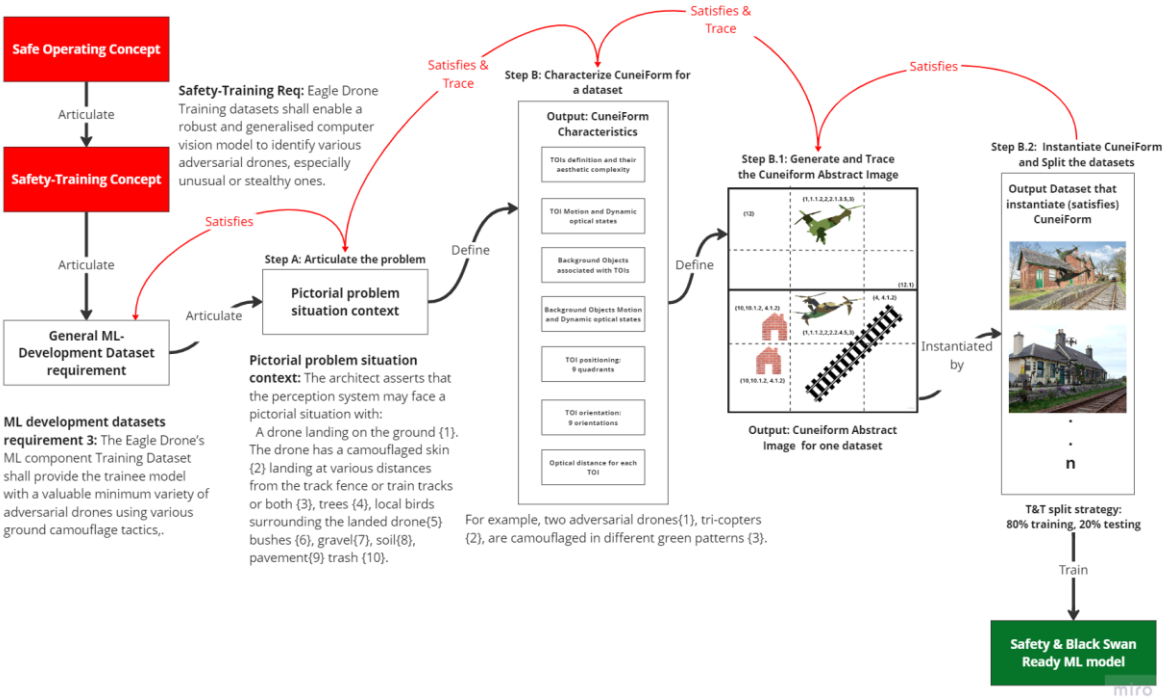


Figure H.35 Cuneiform development process

H.9.1 Step A) Articulate the pictorial problem context:

We start the process by choosing the ML development datasets requirement. In this case study we choose the following:

> **ML development datasets requirement 3:** The Eagle Drone’s ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones using various ground camouflage tactics, such as blending in with bushes, gravel, soil, trash or pavement next to the track zone fence. In order to: train the drone’s machine learning algorithms to dynamically identify, track, and trace adversarial drones obscured by ground-level camouflage.

The above requirement mitigates **ML Safety-Training Requirement 4** from the Safe Operating Concept.

Table H.40 CuneiForm Pictorial situation articulation

Pictorial Situation CoT step	**Definition**
Step 1) Define a minimum variety of TOIs and their pictorial appearances	**Architect prediction:** The architect asserts that the perception system may face a pictorial

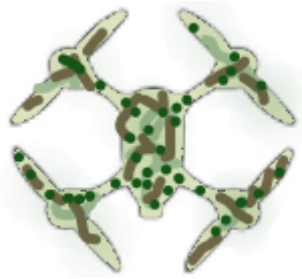
	situation with a drone landing on the ground. The drone has camouflaged skin.
Step 2) Consider a minimum variety of other objects that TOIs aim to influence	**Architect prediction:** The architect asserts that the adversarial drone aims to influence the Eagle Drone detection capability.
Step 3) Consider a minimum variety of objects that TOIs must appreciate	**Architect prediction:** The architect asserts that the adversarial drones may have to appreciate the following environmental **scenery aspects:** distance from the track fence, trees, local birds, bushes, gravel, soil, train tracks, trash and pavement.
Step 4) Consider a minimum variety of what other objects TOIs must control the correctness of predicting their shapes.	**Architect prediction:** The architect asserts that adversarial drones autonomously look to exploit the following aspects: The density of trash on the ground, the density of vegetation on the ground.
Step 5) Produce pictorial problem situation context.	**Pictorial problem context:** The architect asserts that the perception system may face a pictorial situation with: A drone landing on the ground {1}. The drone has a camouflaged skin {2} landing at various distances from the track fence or train tracks or both {3}, trees {4}, local birds surrounding the landed drone{5} bushes {6}, gravel{7}, soil{8}, pavement{9} trash {10}.

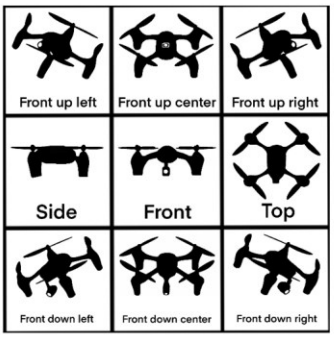
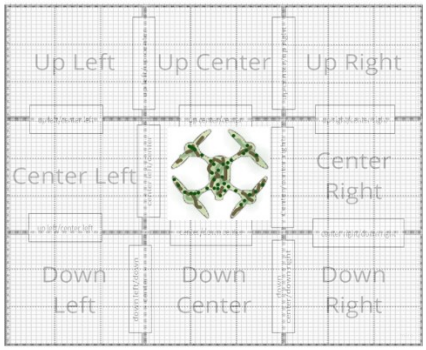
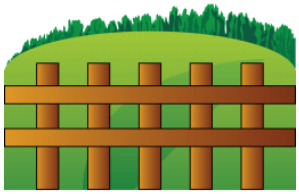
H.9.2 Step B) Characterise the Training Classes for CuneiForms

In this step, we will define one CuneiForm that satisfies the pictorial situation context. We will consider a maximum of 5 objects in this Cuneiform.

Table H.41A Characteristic Training Classes definitions for a CuneiForm abstract image

CuneiForm Characteristic Training Class	**Definition**
Step 1) Define visible horizon attitude.	**At least one horizon attitude type:** **Bird's Eye View:** no horizon is visible

	Frame roll = any°, Frame pitch >-45°{11}
Step 2) Define all the TOIs and their aesthetic complexity. Then, generate abstract representative icons for the CuneiForm abstract image.	At least one adversarial drone{1}, quad-copter type {1.1}, camouflaged in military green pattern {2}. 
Step 3) Define TOI's motion trajectory and dynamic optical situations. Then, update the generated abstract representative icons for the CuneiForm abstract image.	**Motion trajectory:** static, no motion {1.2}. **Dynamic optical situation:** captured without optical blur {1.3}.
Step 4) Define the background objects associated with TOIs and environmental scenery in the background of the CuneiForm. Then, generate abstract representative icons for the CuneiForm abstract image.	Train tracks zone fence or train tracks or both {3}, bushes {6}, gravel{7} trash {10}Train{12}.
Step 5) Define the background Objects' Motion situations and dynamic optical situations. Then, update the generated abstract representative icons for the CuneiForm abstract image.	**Background objects' motion trajectory:** static or in relative motion {3.1,6.1,7.1,10.1,12.1} **Dynamic optical situation:** no motion blur or with blur {3.2,6.2,7.2,10.2,12.2}. Below is an example blurred background object due to relative motion. 
Step 6) Define TOI's positioning in the CuneiForm. Then, generate abstract representative icons for the CuneiForm abstract image.	Center also various {1.4}

Step 7) Define TOI's 3D orientation. Then, update the generated abstract representative icons for the CuneiForm abstract image.	Top {1.5} the following is the definition of an acceptable definition of a minimum variety of possible TOI's 3D orientations  
Step 8) Define the optical distance for each TOI in nindans. Then, update the generated abstract representative icons for the CuneiForm abstract image.	The drone is represented in a pictorial distance of 9 nindans {1.6} bellow is a relative scale of TOI at 9 nindans pictorial distance. 
Step 9) Design the relevant icons to produce the CuneiForm and give an example of an instantiating image	Train tracks zone fence or train tracks or both {3}  bushes{4}  gravel{7}

	 trash {10} 

H.9.3 Final CuneiForm

An output cuneiform would be. An example of an instantiated image is shown in Figure H.36. Can you spot the drone? We define this CuneiForm as a Black Swan Validation set. The training for Black Swan must possess similar characteristics but not be identical.

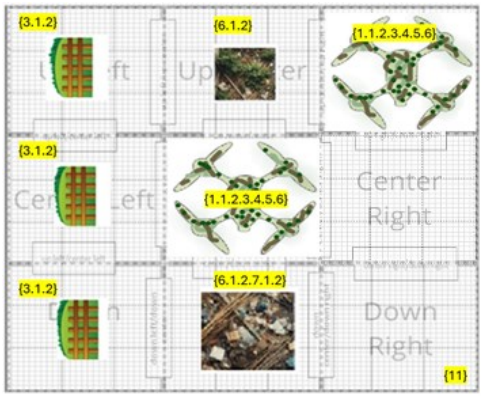


Figure H.36 Example output CuneiForm with appropriate instantiation using the CuneiForm canvas template example.

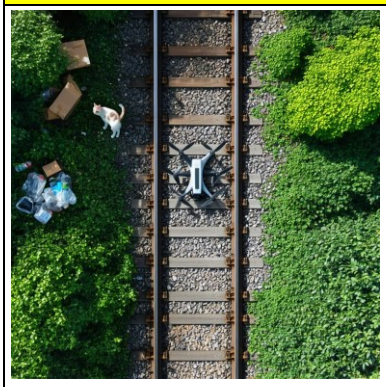
Figure H.36 represents the Cuneiform characterisation for the Black Swan Scenario Validation Dataset, where H.37 is an example. The CuneiForm can be used in the safety case to demonstrate how safety requirements, covering Black Swan Scenarios, have been incorporated into the epistemic training of ML components.

Examples of Out-of-Context (OOC) instantiations:

It is beneficial to provide examples of unsuccessful instantiations of the CuneiForm to help dataset developers refine the accuracy of their data collection process. Additionally, this information will help validators understand the criteria that define accurate instantiations of a CuneiForm, as opposed to those that do not meet these standards. Failure to qualify for a CuneiForm does not mean it to be discarded, but may be considered to be part of a different

CuneiForm. The following table gives an example of what can be an unsuccessful instantiation example for CuneiForm-H.36:

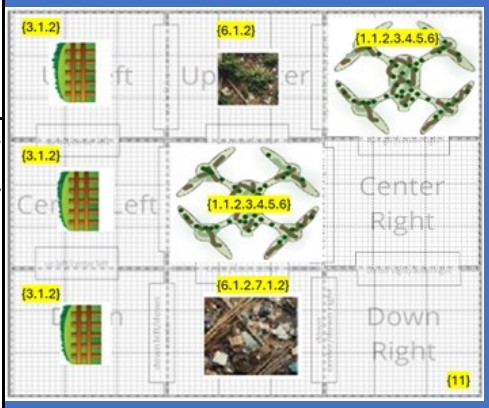
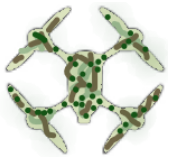
Table H.42A Examples of incomplete or wrong instantiation (OOC) of the output CuneiForm-H.36

Failed Instantized Image	Reasons	Decision
	This image is considered as "non-compliant" to the CuneiForm characterisation. The image does not satisfy the cuneiform because: • Missing the fence. • The drone is not camouflaged. • The drone is not a quadcopter. • There is a cat. However, it does satisfy the following specification: • The drone is at the center. • There is trash in the image. • There are trees, but they can pass as bushes. • There is gravel.	Can be added to the training or validation dataset under different CuneiForm.

The following is the final CuneiForm artefact:

Table H.43B Examples of CuneiForm-H.36 artefact

ML development datasets requirement	**Mitigating ML Safety-Training Requirement 4.** **ML development datasets requirement 3:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones using various ground camouflage tactics, such as blending in with, bushes, gravel, soil, trash or pavement next to track zone fence. In order to: train the drone's machine learning algorithms to dynamically identify, track, and trace adversarial drones obscured by ground-level camouflage.
Pictorial problem	A drone landing on the ground {1}. The drone has a camouflaged skin {2} landing at various distances from the track fence or train tracks or both {3}, trees {4}, local birds

situation context	surrounding the landed drone{5} bushes {6}, gravel{7}, soil{8}, pavement{9} trash {10} Train{12}.	
CuneiForm Training Process Design		
Abstract CuneiForm Characteristics (Training Classes)	**Abstract CuneiForm Training Classes Characteristics definitions: Minimum Valuable Variety**	**Output CuneiForm and an example image**
Visible horizon attitude	**At least one horizon attitude type:** **Bird's Eye View:** no horizon is visible Frame roll = any°, Frame pitch >-45°{11}	CuneiForm Abstract Image: 
TOIs definition and their aesthetic complexity	At least one adversarial drone{1}, quad-copter type {1.1}, camouflaged in military green pattern {2}. 	
TOI Motion and Dynamic optical situations	**Motion trajectory:** static, no motion or appears to be in relative motion{1.2}. **Dynamic optical situation:** captured without optical blur or with blur {1.3}.	Instantiated Concrete Image example: 
Background Objects associated with TOIs	Train tracks zone fence or train tracks or both {3}, bushes {6}, gravel{7} trash {10}Train{12}.	
Background Objects Motion and Dynamic optical situations	**Background objects' motion trajectory:** static {3.1,6.1,7.1,10.1,12.1} **Dynamic optical situation:** no motion blur {3.2,6.2,7.2,10.2,12.2}	
TOI's Pictorial Positioning	Center {1.4} also various	

TOI's 3D Orientation				
			Top(1.5)	
TOI's Pictorial Distance	Both drones are represented in a pictorial distance of 1 nindan (equivalent to 1/9 total area of pictorial frame) {2.5}			

H.9.4 Develop the Training, Testing and Black Swan Validation Datasets

With the CuneiForms produced in the previous step, the ML developer can be tasked with gathering an appropriate number of instantiations constrained by each CuneiForm characterisation. Conventional wisdom can be used to collect and split the training and test. In our case study, we will focus on generating a small-size Black Swan instantiation of the CuneiForm H.36. we will also develop a Black Swan training dataset that would conform to the CuneiForm. The full Black Swan Scenario Dataset can be found in [1].

Black Swan Validation dataset for CuneiForm H.36:

We generate a set of 32 images that capture the CuneiForm. We created the images by manually designing a drone's shell or fuselage and covering it with camouflaged paint. Then, we sought a training track zone where we took pictures near a fence (outside the train track zone) resembling the CuneiForm and what it might look like on the other side of the wall. The following H.37 is an example image from the Black Swan Validation dataset:

Figure H.37A Example of a compliant image to CuneiForm in H.36 in Black Swan Validation set

As you can see, the camouflaged drone landing on foliage is rather hard to detect. We tested a pre-trained model on drones on the Black Swan validation dataset, where the training set had no Black Swan training subset. The result is captured in section H.10.6.4. A model using random drone images performed poorly in a customised operational domain under such Black Swan scenarios. This may indicate that training models on public data is not necessarily reliable for performing well during Black Swan events scenarios.

Black Swan Training dataset for CuneiForm H.36:

Since the model in section H.10.6.4 failed under the Black Swan scenario, we needed to collect the Black Swan training dataset confirming CuneiForm H.36 and re-train the model. We did just that; we gathered 223 images of similar conditions but in different environments. While validation focused on the train track zone environment, the training dataset was obtained from non-train track zone environments, but the CuneiForm characterisation was satisfied with some acceptable variability. Below is a snapshot of what the training Black Swan looked like:

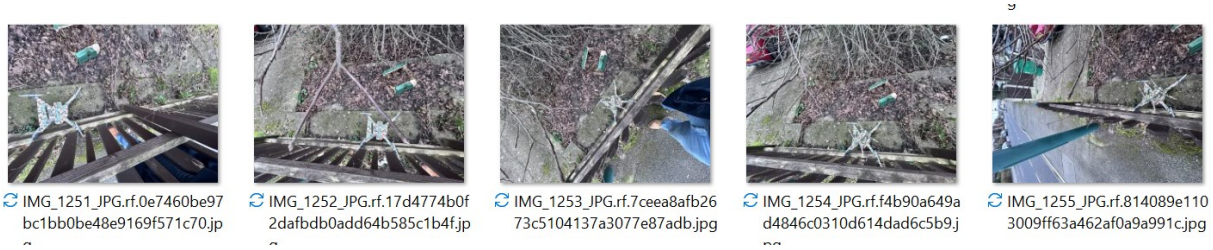


Figure H.38B Example of a compliant image to CuneiForm in H.36 in Black Swan Training Set

H.9.5 More CuneiForms

In this section, we will directly generate CuneiForms to facilitate the development of the dataset. We will refrain from explicitly defining the traceability definitions (the CuneiForm artefact), as the example of cuneiform H.36 is adequate. This section serves solely to illustrate how a set of CuneiForms can fulfil one requirement for a machine learning development dataset. We used those CuneiForms to create further instantiations for a dataset in stage 6.

H.9.5.1 A CuneiForm-based non-typical Black Swan scenarios

We will use those examples to further diversify our Black Swan Scenario. We develop ad hoc cuneiforms for the following criteria:

Table H.44 Set of Black Swan CuneiForms examples

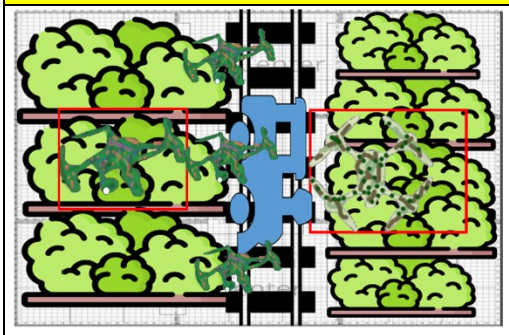
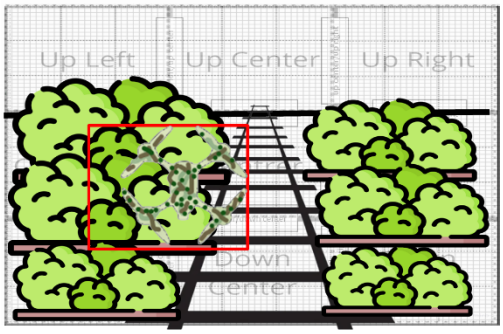
Safety Requirements
ML Safety-Training Requirement 4: Recognise Camouflaged drones on the ground

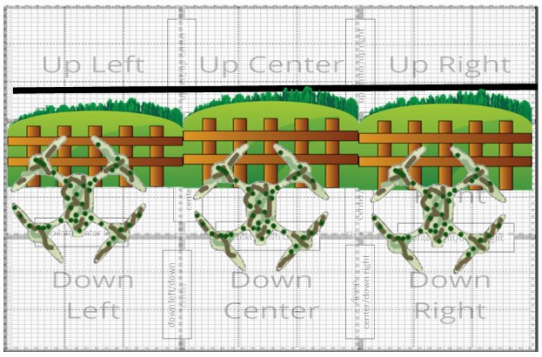
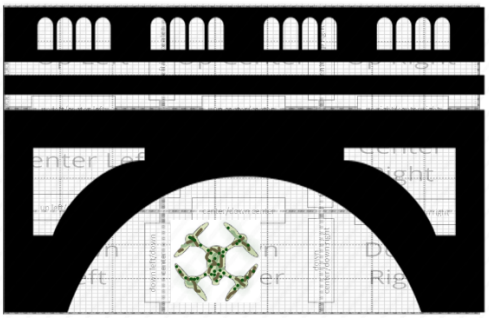
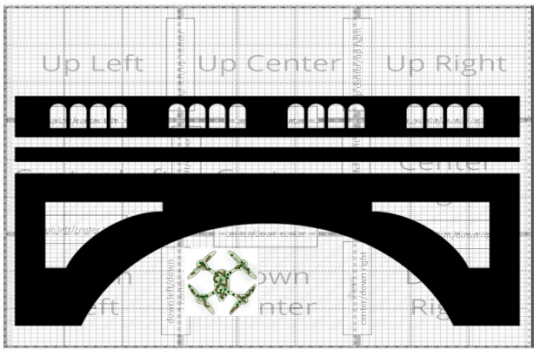
The Eagle Drone's ML component shall be trained to detect, classify, and track adversarial drones using camouflage tactics on the ground by recognising subtle movement patterns, environmental inconsistencies, and spectral differences using AI-driven computer vision models.

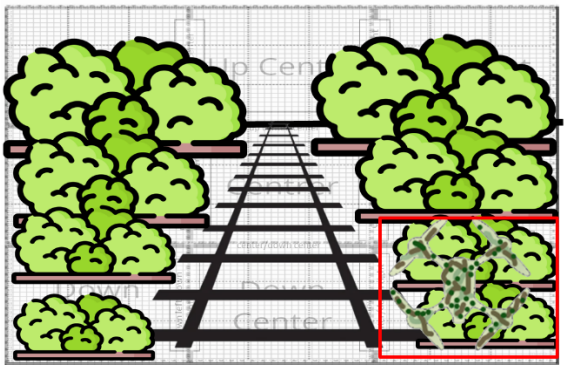
ML development datasets requirement 3:

The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones using various ground camouflage tactics, such as blending in with, bushes, gravel, soil, trash or pavement next to track zone fence.

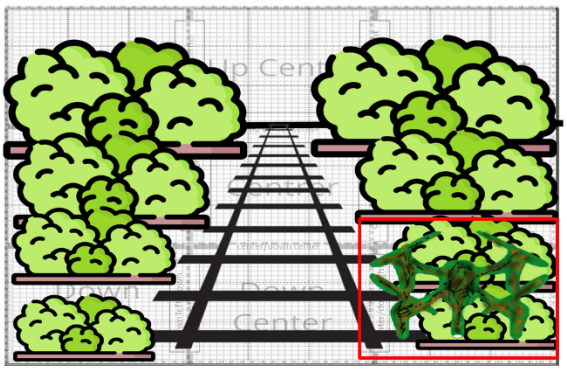
In order to: train the drone's machine learning algorithms to dynamically identify, track, and trace adversarial drones obscured by ground-level camouflage.

Black Swan CuneiForm Scenarios (Batch A)	**Instantiating Image**
 Black Swan CuneiForm H-37 Note, there are several TOIs in the CuneiForm. These are linked by an OR relationship. Meaning we do not expect to see all of them in one image, but rather any of those specific locations.	
 The straight black line represents the Pictorial Horizon Black Swan CuneiForm H-38	

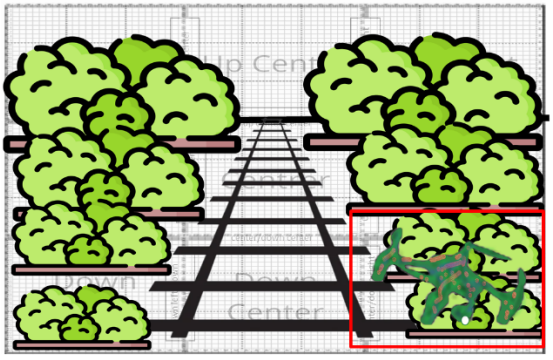
 The straight black line represents the Pictorial Horizon. The TOI can be at any position as depicted. Black Swan CuneiForm H-39	
 Black Swan CuneiForm H-40	
 Black Swan CuneiForm H-41	
Black Swan CuneiForm Scenarios (Batch B)	**Instantiating Image**



Black Swan CuneiForm H-51



Black Swan CuneiForm H-52



Black Swan CuneiForm H-53

In this section, we did not employ the CuneiForm characterisation process; instead, we designed CuneiForms that meet the specified requirements. The rationale behind this decision is to illustrate that it is feasible to create CuneiForms without adhering to the prescribed process. However, regarding the interpretation of the Cuneiform and demonstrating how it effectively meets the established requirements, the architect will need to follow the Cuneiform guidelines and characterisations and produce a report and table that exactly and unambiguously describe how the Cuneiforms are valuable (I.e., traceable to requirements).

H.9.5.2 A CuneiForm-based typical operational scenarios

In this section, we will present a possible training strategy based on carefully selecting which aspects of CuneiForms to focus on for a specific ML development stage. For example, we will choose the TOI 3D orientations and define the split of training, validation and testing strategies.

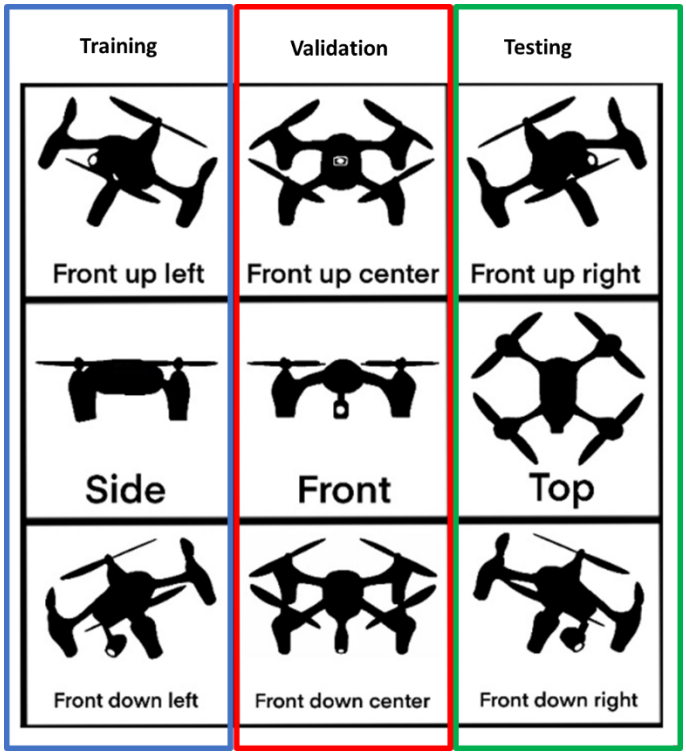


Figure H.39 CuneiForm ML development strategy based on TOI 3D orientation.

Figure H.39 highlights our intended development strategy, to based on training the ML model to recognise specific orientations for the TOI (Front up left, Side, Front down left), then validating it against the following orientation variety (Front up center, Front, Front down center), and finally testing its performance against the following orientations (Front up right, Top, Front down right). For the next batch of datasets, we will fix the following general characteristics of the Cuneiform and only alter the orientations of the 3D orientation.

The CuneiForm strategy will be based on mitigating the following safety and dataset development requirements:

Table H.45A Dataset requirement chosen example.

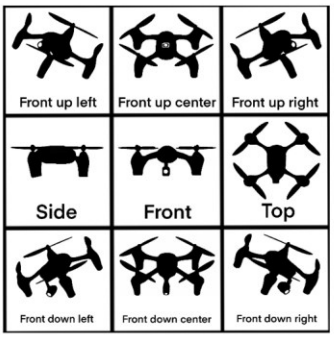
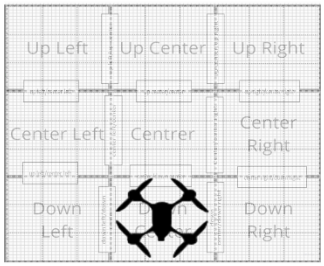
ML Safety-Training Requirements (Training Concept)	**ML Safety-Training Dataset Requirement**
ML Safety-Training Requirement 2: The Eagle Drone's ML component shall be trained to recognise adversarial drones	**ML development datasets requirement 2:**The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial

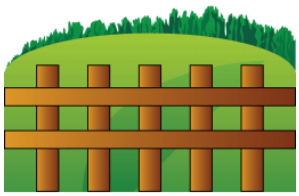
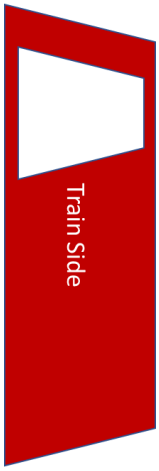
approaching the train tracks and during their landing by the tracks side.	drones (typical off-the-shelf drones) approaching the train, train tracks and attempting to land next to the train tracks boundary. **In order to:** train the drone's machine learning algorithms to dynamically identify, track, and trace adversarial drones during typical operations.

The following is a CuneiForm characterisation for the training strategy over the common operational domain on a train track:

Table H.46B Typical operations CuneiForm characterisation.

CuneiForm Characteristic Training Class	**Definition**
Step 1) Define visible horizon attitude.	**Elevated Level Horizon:** Frame roll = −1≤ROLL≤1 Frame pitch = 1<PITCH≤45 **Also,** various elevated horizon attitudes are acceptable {1}
Step 2) Define all the TOIs and their aesthetic complexity. Then, generate abstract representative icons for the CuneiForm abstract image.	At least one adversarial drone {2} 
Step 3) Define TOI's motion trajectory and dynamic optical situations. Then, update the generated abstract representative icons for the CuneiForm abstract image.	**Motion trajectory:** static, no motion {2.1}. **Dynamic optical situation:** captured without optical blur {2.2}.
Step 4) Define the background objects associated with TOIs and environmental scenery in the background of the CuneiForm.	Train tracks zone or train tracks or both{3}, bushes and trees {4}, Side of the Train {5}.

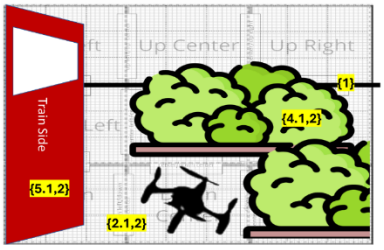
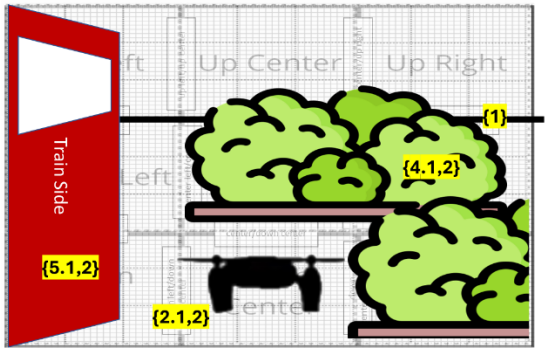
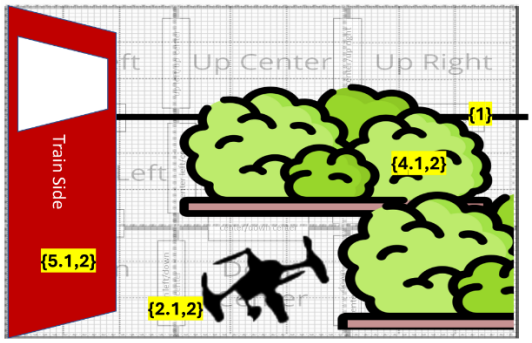
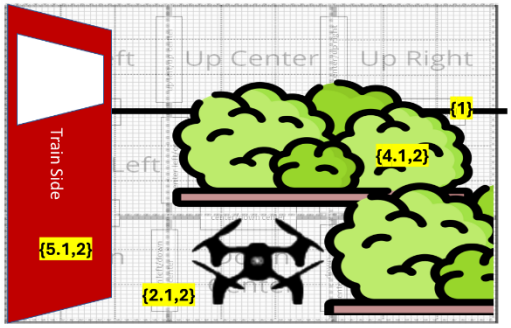
Then, generate abstract representative icons for the CuneiForm abstract image.	
Step 5) Define the background Objects' Motion situations and dynamic optical situations. Then, update the generated abstract representative icons for the CuneiForm abstract image.	**Background objects' motion trajectory:** static or in relative motion {3.1,4.1,5.1} **Dynamic optical situation:** no motion blur or with motion blur {3.2,4.2,5.2}
Step 6) Define TOI's positioning in the CuneiForm. Then, generate abstract representative icons for the CuneiForm abstract image.	Down Center {1.4}
Step 7) Define TOI's 3D orientation. Then, update the generated abstract representative icons for the CuneiForm abstract image.	Training TOI at (Front up left, Side, Front down left), validation TOI at (Front up center, Front, Front down center), testing TOI at (Front up right, Top, Front down right){1.5} the following is the definition of an acceptable definition of a minimum variety of possible TOI's 3D orientations  
Step 8) Define the optical distance for each TOI in nindans. Then, update the generated abstract representative icons for the CuneiForm abstract image.	The drone is represented in a pictorial distance of at least 9 nindans {1.6} bellow is a relative scale of TOI at 9 nindans pictorial distance. 

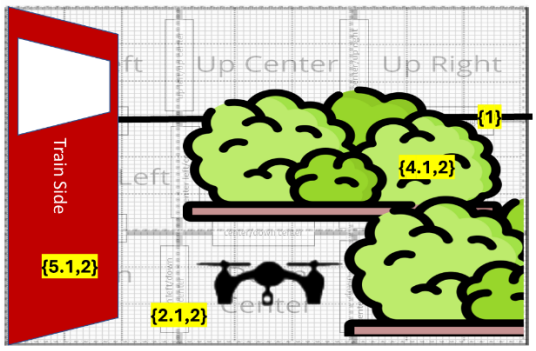
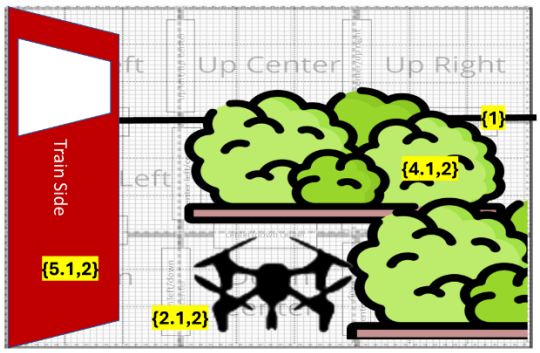
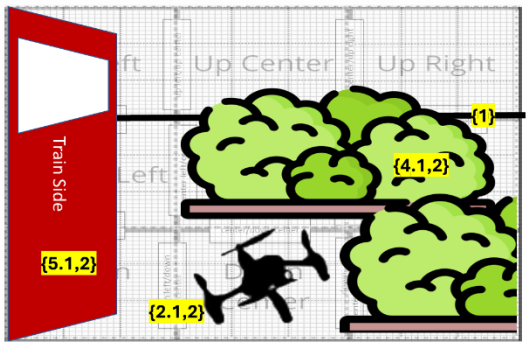
Step 9) Design the relevant icons to produce the CuneiForm and give an example of an instantiating image	Train tracks or train tracks or both {2}  bushes{3}  Side of train{4} 

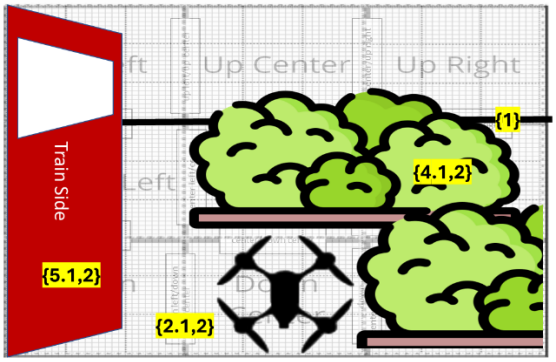
The following is a set of CuneiForms that enact the above requirements: typical drone and typical operational behaviour i.e. no black swan events:

Table H.47 Set of Black Swan CuneiForms examples

Safety Requirements	
ML Safety-Training Requirement 2: The Eagle Drone's ML component shall be trained to recognise adversarial drones approaching the train tracks and during their landing by the tracks side. **ML development datasets requirement 1:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drone shapes as they approach the track zone boundary fence and airspace from different angles and altitudes, including both direct and evasive approaches.	
CuneiForm-based Training Scenarios	**Instantiating Image** (Total 1000 similar, not exactly the same, images generated)

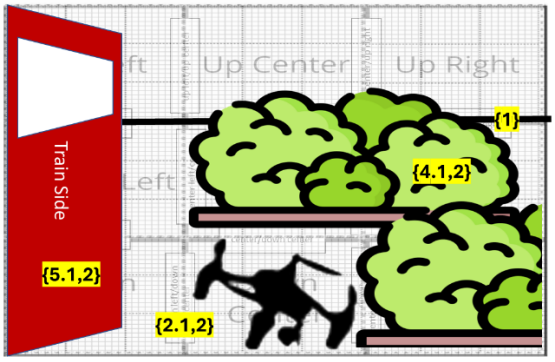
Common Operations CuneiForm H-42  TOI at 3D orientation of: Front up left	
Common Operations CuneiForm H-43  TOI at 3D orientation of: side	
Common Operations CuneiForm H-44  TOI at 3D orientation of: Front down left	
CuneiForm-based Validation Scenarios	**Instantiating Image**
Common Operations CuneiForm H-45  TOI at 3D orientation of: Front up center	

Common Operations CuneiForm H-46  TOI at 3D orientation of: Front	
Common Operations CuneiForm H-47  TOI at 3D orientation of: Front down center	
CuneiForm-based Testing Scenarios	**Instantiating Image**
Common Operations CuneiForm H-48  TOI at 3D orientation of: Front up right	
Common Operations CuneiForm H-49	



TOI at 3D orientation of: Top

Common Operations CuneiForm H-50



TOI at 3D orientation of: Front down right

H.10 Black Swan Datasets Experiment Description and Analysis[8]

In this section, we attempted to examine how the generated Black Swan assists with constructing context-aware datasets. We conducted several experiments where we tested model performance on different training, validation and testing strategies. The experiments aimed to evaluate how context-aware dataset construction, particularly the inclusion of Black Swan (non-typical, Out-of-Distribution (OOD) scenarios, impacts the robustness and trustworthiness of object detection models. A secondary goal was to assess whether testing on Black Swan scenarios strengthens assurance arguments for model reliability in safety-critical applications.

H.10.1 Prior experimentations

A significant body of academic research highlights that evaluating machine learning (ML) models solely on In-Distribution (InD) test data, data drawn from the same distribution as the training set, can lead to misleadingly optimistic assessments of model performance. This issue is particularly critical in safety-sensitive applications, such as drone detection systems for securing train track zones, where models must reliably handle OOD scenarios, including adversarial drone intrusions.

Deep learning models often exhibit overconfident predictions when evaluated solely on InD test sets, undermining the reliability of their confidence scores in real-world applications. Ovadia et al. [2] conducted a comprehensive empirical study, showing that modern neural networks typically become poorly calibrated under dataset shifts. Models that appear confident on InD data yield misleading uncertainty estimates when presented with OOD inputs, a critical failure mode for safety-sensitive systems. Building on this, Cai et al. [3] proposed the Distribution Shift Decomposition (DISDE) framework to quantify how performance degrades under explicit distributional changes, demonstrating that high InD accuracy does not guarantee robustness to OOD scenarios. Similarly, Taori et al. [4]investigated ImageNet classifiers. They found that robustness to synthetic perturbations (e.g., random noise or blur) does not necessarily translate to resilience against natural distribution shifts encountered in deployment. Collectively, these studies highlight that relying solely on confidence scores derived from InD evaluations can lead to unwarranted trust in model outputs; instead, systematic evaluation under distributional shifts is necessary to ensure that uncertainty estimates remain meaningful in practice.

H.10.2 Why are we experimenting

Although prior work has clearly demonstrated that neural networks evaluated solely on InD test data can exhibit overconfidence and provide misleadingly optimistic performance estimates

[8] For more details see section H.10

under distributional shift, it remains equally important to assess how these insights translate into concrete decision-making scenarios in safety-critical domains. The prior work provides further justification for designing an ML development pipeline that clearly addresses OOD scenarios and reinforces our understanding of the composition of a valuable test dataset, which we trust to inform any performance metric.

Suppose a regulator or quality-assurance engineer must choose between two models with differing data exposures: Model 1, trained on a large corpus (85,677 images) and achieving 93% test accuracy, and Model 2, trained on a smaller corpus (16,788 images) and achieving 99% accuracy. The literature underscores that higher InD accuracy does not guarantee robustness to OOD inputs; however, practical safety-case decisions often hinge on intuitive judgments about "experience" versus "peak performance". By embedding this phenomenon into a thought experiment, we concretise the abstract concern of calibration under shift and force an explicit consideration of whether dataset breadth (and hence diverse exposure) should outweigh nominal test-set accuracy.

In doing so, we highlight how reliance on InD metrics alone may lead regulators to sign off on models whose apparent excellence masks brittleness in rare or adversarial scenarios—and how invoking regulatory/legal risk motivates a more rigorous evaluation of uncertainty, model calibration, and the trade-offs between dataset size, contextual relevance, and safety assurances.

H.10.3 Thought Experiment

Before we start the experiment phase, we would like you to consider answering the following question;

Imagine you are a Safety Critical AI regulator or quality assurance engineer. You are in charge of signing off on the safety of an autonomous system in a safety-critical application involving detecting drones. The ML engineering team provided you with two ML models that are tasked to detect drones reliably.

- **Model 1:** Trained on 85677 images and achieved 93% success on the test dataset.
- **Model 2:** Trained on 16788 images and achieved 99% success on the test dataset.

Considering the size of the training dataset alone as the main factor to judge (assuming all other factors have been accounted for from a safety design perspective), which model would you trust more and therefore be willing to sign your name on it? Bear in mind the consequences of releasing unsafe AI could lead to legal persecution.

Please write your answer somewhere and save it before looking at the experiments below.

Our answer to that would be Model 1, because although it scored slightly less, the model has more experience and exposure to a variety of examples, which means 93% performance is more trustworthy than Model 2, which has 99% success rate.

H.10.4 Architect ML training strategy dilemma

Usually, the popular wisdom about the quality of training of an ML model boils down to a simple approach to training:

Feed as much data as possible into the training. Online datasets are cheap, so gather all that you can get from the internet. But! they are context-unaware datasets.

In practice, we faced a dilemma when we wanted to gather data at this stage. Should we obtain a large but diverse dataset with unknown biases (difficult to verify), or a small but context-aware dataset?

The intuition above is helpful when we must justify subjective engineering decisions to favour size or distribution. Like for example, a possible trade-off would be:

- Should the architect accept a training syllabus that produces a very large, cheap, context-unaware dataset but with a higher likelihood of biases?
- Or a smaller choice, but context-aware dataset and assured balance.

The safety case needs to answer this question because, in either situation (where there could be no right answer), some rigour is required to justify the claim of choosing which one. A biased but larger dataset, or a balanced but smaller dataset. Therefore, we need a more rigorous metric that facilitates objective decision-making to support safety claims when an architect opts not to include relevant images. In the event of a catastrophic failure, if regulators discover that the architect had additional data that was omitted from the dataset, this could lead to questions of negligence.

On the other hand, if more data were included but the overall dataset exhibited a biased distribution, it would raise the question of why a more balanced, albeit smaller, dataset was not prioritised. In this section, we will explore the concept of uncertainty as it directly pertains to addressing this dilemma and the potential legal inquiries that may arise following catastrophic failures in complex systems.

H.10.5 Methodology

The experiments were designed to evaluate the impact of context-aware dataset construction on object detection models' robustness, particularly when exposed to Black Swan scenarios (non-

typical, high-impact events). The methodology focused on testing the hypothesis that **explicit** inclusion of Black Swan examples in training improves model performance on rare operational scenarios and strengthens assurance arguments for safety-critical systems.

H.10.5.1 Hypothesis

Hypothesis 1 (Hyp1): A model trained and validated primarily on typical operations or OOC images (context-unaware datasets) may perform well on similar "typical" validation/test data, giving a false sense of security. Still, it may fail dramatically on "black swan" (non-typical scenarios, OOD) images unless those black swan scenarios are predicted and represented in training. In other words:

1. **Black Swan Context aware dataset:** Including in-context black swan examples in the training set is essential for better performance and more convincing assurance arguments on black swan test data.

2. Furthermore, testing over a dedicated Black Swan scenarios test set is a valuable metric to ensure that the trained model is ready to handle rare and high-impact scenarios.

3. To demonstrate that a dataset is a Black Swan, we need to show the following properties:

    a. The engineered dataset is OOD relative to training + validation.

    b. The trained model performs poorly.

    c. It is in-context of a Black Swan scenario.

Specifically, we ask:

Q1: Does a model trained on only typical or OOC images perform better or poorly on black-swan scenarios, and, conversely, does explicitly including black-swan images in training measurably improve the model's performance on black-swan test sets?

These questions underpin the motivation to systematically compare the performance of 9 different training-validation-testing strategies under two core experiment types:

- Testing over Black Swan robustness evaluation:

    - **Expected results:** if trained on typical and out/context, the model may experience noticeable performance reduction over Black Swan test datasets, unless trained on similar Black Swans.

- **False sense of security:** Testing over typical in-context operational performance evaluation gives a false sense of security if trained without Black Swans training:

- **Expected results:** A misleading measure of good performance, as they may not perform as intended in Black Swan events.

- **Validating Black Swan distribution** demonstrates that a test dataset is a Black Swan data shift relative to training and validation; we must demo that the testing set is a shifted dataset from the training and validation. This means we need to show that we can produce OOD datasets.

Model Architecture and Datasets

- **Model**: All experiments used the **Roboflow 3.0 Object Detection (Fast)** architecture with **COCO-trained weights** as initial checkpoints. No image augmentations were applied to isolate the impact of dataset composition.
- **Datasets**:
  - **In-context Black Swan scenarios**: Generated syntactically and guided by CuneiForms (structured operational scenarios from Stage 4).
  - **OOC images**: Sourced from five drone-centric datasets (e.g., DRONES_NEW, Drone Detection) to simulate generic backgrounds.
  - **Typical operational scenarios**: In-context images representing standard use cases, generated syntactically and guided by CuneiForms.

Key Assumptions

1. **mAP@50 as the primary metric**, aligning with an IoU threshold of 0.5 for deployment.
2. **Dataset sufficiency**: Assumed adequate size for all models to generalise, despite variability in composition.

H.10.5.2 Scope Limitations

The current experiments do not address:

- Sensitivity to threshold variations for detection confidence and bounding box overlap.
- Real-world, dynamically acquired operational data. Only realistic synthetic data (overlaying drones over real-world footage).
- Effects of image pre-processing or data augmentation techniques.

These are planned future extensions of this work to further validate and refine the insights gained from this study. Overall, this experiment design is a foundation for evaluating how context-aware data curation strategies influence ML model robustness, particularly in safety-critical domains where both typical and unforeseen (Black Swan) scenarios are of concern.

H.10.5.3 Experiment Setup

Nine experiments were conducted, grouped into three categories to isolate the effects of dataset composition:

Table **Error! No text of specified style in document.**.48 Experiments grouping

Group	Description	Key Experiments
Group 1	Training on out/context images only. [context-unaware training and validation datasets]	Exp. 1, 4, 5
Group 2	Training on in-context typical operations + out/context images. [typical operations context-aware training and validation datasets]	Exp. 6,7,10
Group 3	Training on in-context Black Swans + out/context images. [Black Swan context-aware training and validation datasets]	Exp. 2, 3, 8.1,8.2
Group 4	Training in all types [Black Swan, Typical ops context-aware training and validation datasets]	Exp. 9.1,9.2,9.3,9.4. 9.5

Experiment Variables

- **Training/Validation/Test Splits**: Varied proportions of in-context Black Swan, typical, and OOC images.
    - Example: Exp. 3 used **30% Black Swan images** in training, while Exp. 4 used **100% OOC images**.
- **CuneiForm Integration**:
    - **Black Swans**: Batch A (H.36–41) and Batch B (H.51–53)[9].
    - **Typical operations:** H.42, 43,44, 45,46,47,48,48,50[10].
    - **OOC:** H.54.
- **Pre-processing**: Limited to resizing (320x320 in Exp. 5/10) to avoid confounding effects.

Example Experiment Configurations:

For a full summary, see section H.10.5. Remember that when we say “trained”, we also include validation, since it is a critical training component. For example:

- **Experiment 1**: Trained and tested on 100% OOC images.
- **Experiment 3**: Trained and validated on 30% Black Swan + 70% OOC images; tested on 100% Black Swan Batch A CuneiForms.

[9] See Table 6.26 Set of Black Swan CuneiForms examples.
[10] See Table 6.29 Set of typical operations CuneiForms.

- **Experiment 10**: Combined (3270 images) 4% in-context typical operations with (82257 images) 96% OOC images; tested on 100% Black Swan Batch A.

H.10.5.4 Testing Approach

Performance Metrics

- **mAP@50**: Primary metric for validation/testing.

Visual Similarity Testing: To validate dataset independence /comparability (across experiments) and Black Swan "OOD" status:

- **Perceptual Hashing (pHash)**: Computed unique visual signatures for each image using the imagehash library.
- **Venn Diagrams**: Quantified overlap between datasets using unique pHash values.
    - **IoU Metric**: The Jaccard similarity coefficient, **J(A,B)**, is given by:

$$J(A,B) = \frac{|Intersection|}{|Unique\ pHash\ count\ A| + |Unique\ pHash\ count\ B| - |Intersection|}$$

Example 1: Exp. 4's Black Swan test set had **0% IoU** with training data (Figure 6.32), confirming full OOD status. In this experiment, the model trained on Exp.4 training+validation dataset performed 24% on the OOD Black Swan exp4_testing dataset. demonstrating that exp4_testing is a Black Swan data shift in comparison to exp4_training+validation.

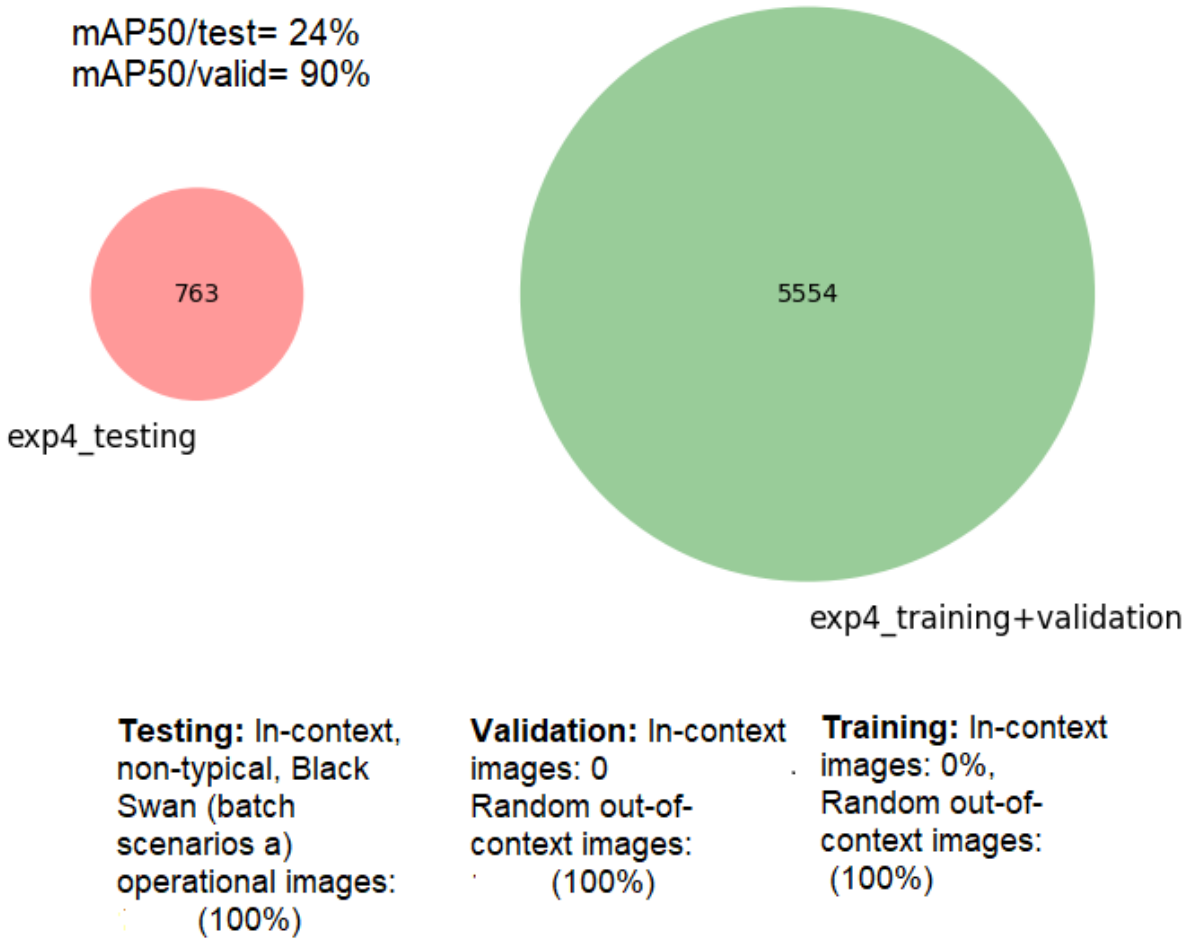


Figure **Error! No text of specified style in document.**.40 similarity diagram between training and testing datasets in exp.4

Example 2: Exp. 1's test set has 10**% IoU** with training+validation data, confirming partial OOD status. Which means there are 576 unique p-hashes which are shared in both datasets. 278 unique p-hashes only exist in the test, and there are 4475 unique hashes in the training and validation combined. Both datasets are composed of only OOC datasets.

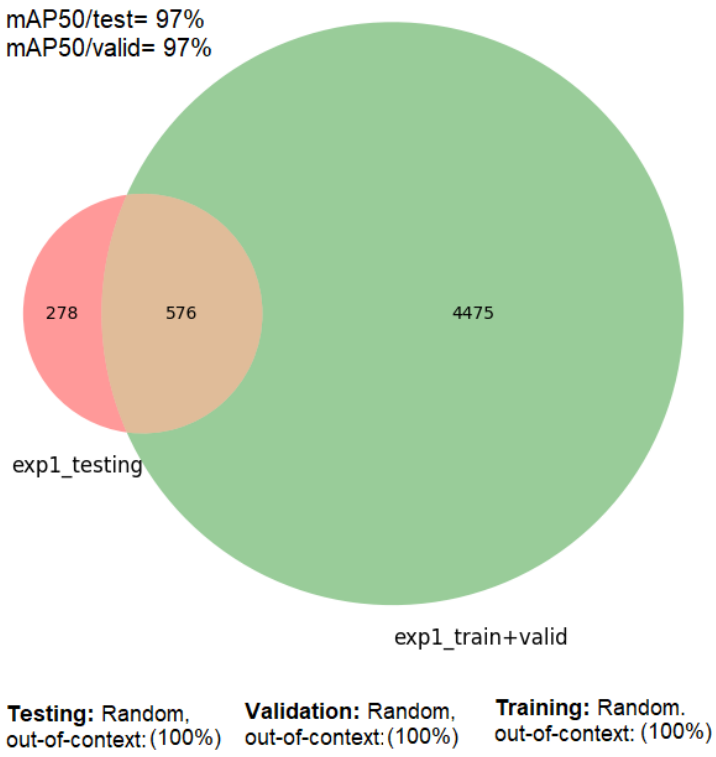


Figure **Error! No text of specified style in document.**.41 Exp1 Similarity test

To see how comparable Exp1 to Exp.4 we compared the similarity between the training of each exp. We also produced a Python code that gives a similarity test report between any two datasets. For example:

exp1_train+valid dataset: 5051 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of approximately 65.60%.

exp4_train+valid dataset: 5554 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of roughly 63.50%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 4891 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets **IoU**) between exp1_train+valid and exp4_train+valid: 85.59% **using J(A,B) formula.**

comparing dataset without in-dataset similarities (considering only one hash number if repeated multiple times):

exp1_train+valid -> exp4_train+valid: (4891 / 5051) ≈ 96.83%.

exp4_train+valid -> exp1_train+valid: (4891 / 5554) ≈ 88.06%.

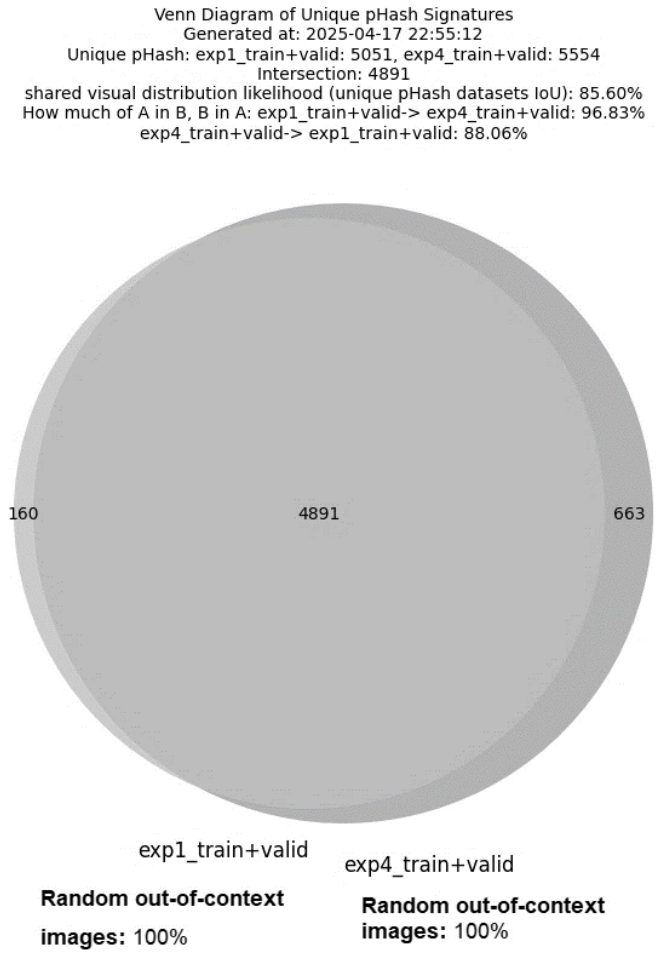


Figure **Error! No text of specified style in document.**.42 Comparing the training syllabus in Exp.1 and 4

The output of the similarity test includes details such as:

- The names of the images which share the identical hashes between datasets.
- The names of images that exist only in each dataset individually.

Figure 6.34 tells us that the trained model should have comparable results. Yet, although the model trained on Exp.1 training + validation strategy (only OOC data) gave us a sense of security that it can succeed 97% of the time when it observes a drone, the in-context Black Swan test revealed the total opposite. It showed that the model is unreliable when exposed to Black Swan data shifts. Thus, the approach in Exp.1 gave a false sense of security and that Black Swan OOD is a very good quality check. Suppose the model can be reliable during typical day-to-day operations and handle Black Swan rare changes in operational conditions (captured in Black Swan datasets). In that case, it is a model that can give us a better sense of security about its reliability and performance.

3. Key Comparisons

- **Within-Group**: Contrasted Exp. 1 (out/context-only) with Exp. 4 (Black Swan testing).

- **Cross-Group**: Compared Group 3 (Black Swan-trained) with Group 1/2 to evaluate robustness gains.

H.10.5.5 Experiment Results

Results were grouped to highlight performance trends on Black Swan and typical test sets:

Table **Error! No text of specified style in document.**.49 Experiments results[11]

Group	Experiment	Training Size	Train + Validation strategy			Test / mAP@50		
			out/ context	Black Swan	Typical ops	Black Swan	Typical Ops	Out/ context
1	Exp. 1	6753	yes	0	0	-	-	97%
	Exp. 4	8000	yes	0	0	A: 24%	-	-
	Exp. 5	9924	yes	0	0	-	93%	-
2	Exp.6	10015	yes	0	yes	-	99%	-
	Exp.7	10015	yes	0	yes	A: 13%	-	-
	Exp. 10	85677	yes	0	yes	A: 35%	-	-
3	Exp. 2	6753	yes	**Train:** A **Valid:** 0	0	-	-	97%
	Exp. 3	8000	yes	A	0	A: 67%	-	-
	Exp. 8.1	8000	yes	A	0	-	97%	-
	Exp. 8.2	8000	yes	A	0	B: 28%	-	-
	Exp. 9.1	14961	yes	A	yes	-	Both: 99%	
	Exp. 9.2	17196	yes	A	yes	B: 40%	-	-
4	Exp. 9.3 (augmented)	51588	yes	A	yes	B: 44%	-	-
	Exp. 9.4	16788	yes	A+B	yes	All:99%		
	Exp. 9.5	16788	yes	A+B	yes	A+B:98%	-	-

H.10.5.6 Key Insights

The experiments conducted revealed some insights about the challenges of training models on typical operations or OOC datasets without incorporating Black Swan scenarios. Despite increasing the size of OOC datasets, models demonstrated poor performance during Black Swan events, highlighting a critical weakness in training syllabus. Key lessons learned include the necessity of including diverse and challenging OOD data to improve model reliability. Specifically, the absence of Black Swan training data can lead to a false sense of security, as seen when models trained solely on typical operations struggled to handle rare scenarios effectively.

[11] See Table H.55 for a detailed description of the results and reference to online repositories.

The findings emphasise the importance of systematically identifying and incorporating Black Swan cases into training to demonstrate models' ability to adapt to unforeseen data shifts. However, demonstrating the deliberate inclusion of Black Swans requires validation that the dataset comprises relatively unforeseen scenarios. To achieve this, we could use dissimilarity measures to show that a dataset is OOD and that the current model performs poorly on it. This can be illustrated in the safety case where Black Swans were systematically identified and incorporated into training.

The following is a summary table that captures what we learned from our experiments on the structure of dataset context awareness.

Table **Error! No text of specified style in document.**.50 Interpretation of results and key insights

Key insight	**Elaboration**	**So what? lessons learned**
No Black Swans in training, poor performance during Black Swan scenarios.	Exp 4,7, and 10 showed poor performance during Black Swans despite the systematic increase in OOC dataset size. Especially 10, we injected in-context typical operations images (3270 images), the performance did not significantly increase over in-context Black Swan (scenarios A).	**Lesson 1:** No matter how much OOC data we use to train the model, it may not be sufficient to assure performance during Black Swan scenarios (edge cases, as they are commonly known). If we cannot trust the model during a Black Swan situation, why should we trust it at all?
	This was also observed during Exp 3, which had significantly less size training (nearly 10 times less) than Exp 10. The presence of Black Swan scenarios in the training and validation led to a notable increase in performance over Black Swan scenarios A.	**Lesson 2:** Black Swan testing revealed a vulnerability in the training syllabus. This is a useful finding for assurance. **Lesson 3:** The Stage 4 process helped us predict a challenging scenario. If we relied solely on OOC data, we would have missed such a data shift. This lesson is also reaffirmed in Exp 8.1.

	in Exp 8.2 (which used the same training datasets as in 3 and 8.1) although we included Black Swans A, performance over OOD Black Swans B also dropped. Which further confirms that the absence of deliberate black swans does not assure performance during a given Black Swan operation.	Black Swans A and B helped us note the epistemic gap of knowledge in our training syllabus. When we trained the model on Black Swans A, we expected similar performance over B scenarios. However, this did not happen. **Lesson 4:** We need to systematically predict OOD black swans, test an earlier version of the model to confirm they are challenging and then reintroduce them into the training. This way we can be assured that we did consider Black Swans.
False sense of security over Out/context and typical ops	Exp 6 used the same training set as experiment 7; the model that was trained on out/context + typical ops showed a 99% success rate over typical ops testing. But the same model in Exp 7 showed a significant drop in performance during Black Swan scenarios A (only 13%). The same behaviour was observed in Exp 1 and 4. **Although** there is a **notable** increase in OOC **training** dataset in comparison to Exp 1, the trained model dropped from performing 97% (over out/context testing), to only	**Lesson 5:** Along with, Exp 1, 2 and 9.1, testing on typical operations and OOC, whether training had Black Swan, or not, whether it had typical ops and OOC, testing on typical operations and/or OOC datasets may give a false sense of security. Black Swan testing (OOD datasets) helps us train the model to handle potential unforeseen data shifts and rare contexts in the operational domain.

	24% over Black Swan testing. This shows that Training and testing.	
Black Swan-driven ML development pipeline: Discovering a Black Swan requires demonstrating that a Black Swan has been caught and processed to provide assurance.	Two factors allowed us to note the Black Swan data shift: 1) The similarity test revealed an OOD relative to the training set. 2) Poor performance of the trained model against the OOD dataset. Those two factors inspired us to understand that we need a systematic process that starts with a reference training set (like typical ops) and then tests Black Swans datasets against the trained model over such a reference dataset. We noticed an interesting behaviour related to Exp.5. Exp 4, 7, 8.2, 9.2, and 10 all share one aspect: the test set is completely OOD with the train and validation sets. The trained model has performed poorly. Exp5 also share the same OOD case. However, the model performed relatively well. This demonstrate that the OOD of exp5 test set may mitigated by	**Lesson 6:** to demonstrate a systematic training for Black Swan data shifts we need: 1) Predict Black Swan scenarios. 2) Generate CuneiForms. 3) Generate an initial reference dataset that is not Black Swan. 4) Generate Black Swan images. 5) Perform Similarity tests to confirm OOD status. 6) Train the model on the reference dataset (not Black Swan). Let's call it the White Swan model. 7) Test the White Swan model over Black Swan. 8) If it fails, then the Black Swan is an unmitigated black Swan, and that is good news (we caught one!).. Now, we include it in the training pile. 9) If it passes, then the dataset's diversity is good enough to mitigate such a Black Swan data shift.

	the diversity of the training and validation. This means not all OOD data is are **challenge** for a trained **model**. This means we need to test whether the model fails in an OOD scenario before we judge whether it is mitigated or not.	This process underpins our approach to Black Swan-led ML development and assurance.
Augmentation did not help much with Black Swan scenarios	Exp9.3 showed that augmenting the dataset would not significantly increase reliability if we had a training dataset that did not perform well against a Black Swan scenario.	**Lesson 7:** Use augmentation techniques as a bonus after ensuring the model is trained and Black Swan tested. Achieving high performance over epistemic coverage in training and validation datasets is more important than using augmentation.
Unmitigated and mitigated Black Swan data shifts	Data shift is a real **challenge** for the reliability of ML components [5]. From a safety assurance perspective, predicting them and then validating them in a pictorial sense is also a challenge. Experiments 4, 10, 8.2 and 9.2, with the help of stage 4 and similarity measures, helped us to define the data shift, which constitutes a Black Swan scenario. We were under the impression that if we tested ML model over OOD yield poor performance.	**Lesson 8: Combining** the realisation in exp 4,10,8.2, 9.2 and 5, we learned that: A data shift is an OOD dataset relative to another. Some data shifts are mitigated and can be validated by good performance, while others are not mitigated (validated by poor performance). We explain that nature of the epistemic diversity in training and validation is an influential factor on whetehr a data shift is mitigated or not. This means

	However, the idea was challenged when we conducted exp.5 in which the trained model (trained with OOC only dataset) performed well over in-context typical operations, which **were** OOD with the training and validation. It means that being OOD is not enough to tell whether a dataset is a Black Swan. We also need to see if a model is performing poorly. This means that some OOD datasets are mitigated by the diversity of the training and validation, while others are not. Hence, we use the term "mitigated" or "unmitigated". Similarity measures help validate the visual data shift among datasets.	we can validate such process by simply: 1) Define a reference dataset. 2) Define a targeted data shift scenaros and images. 3) Compare the targeted datset similarity to the reference dataset. 4) train a model over the reference dataset. 5) If the model **performs** poorly, **then the data shift in the** targeted dataset is not mitigated by the epistemic diversity of the reference dataset.
Ghost patterns	Experiments 5 and 6 demonstrated that an invisible set of pictures is part of the training and validation (evidenced by comparable ML performance despite being OOD). We can reasonably trust that safety properties emerge during such scenarios even when they do not exist in the training. We call these	**Lesson 8:** based on whether Black Swans are mitigated or not mitigated, we can understand that practically**,** There are two types of invisible pictorial sets: 1. scenarios where the model is expected to perform well over them (mitigated black swans), but does not existing in

	Friendly White Ghost Patterns (pictorial scenarios). If such scenarios exist, then the opposite of them also exists. These are the Black Ghost scenarios, which are not part of the development set, and safety properties can be guaranteed to emerge.	any development set, yet we expect safety properties to be preserved in them. We call them White Ghost patterns. 2. Scenarios that are not part of the invisible White Ghost patterns, or any other development set where we expect that safety properties may not emerge as expected (unmitigated black swans). Those are Black Ghost patterns. This PhD will only mention their presence, but will not investigate them further as they will be a future research endeavour.

If we were to compare the training syllabus between Black Swan-experienced and inexperienced models, 9.4 and 10, we would realise clearly that size does not matter in terms of trust in performance over Black Swan scenarios. See the following similarity test report between 9.4 and 10:

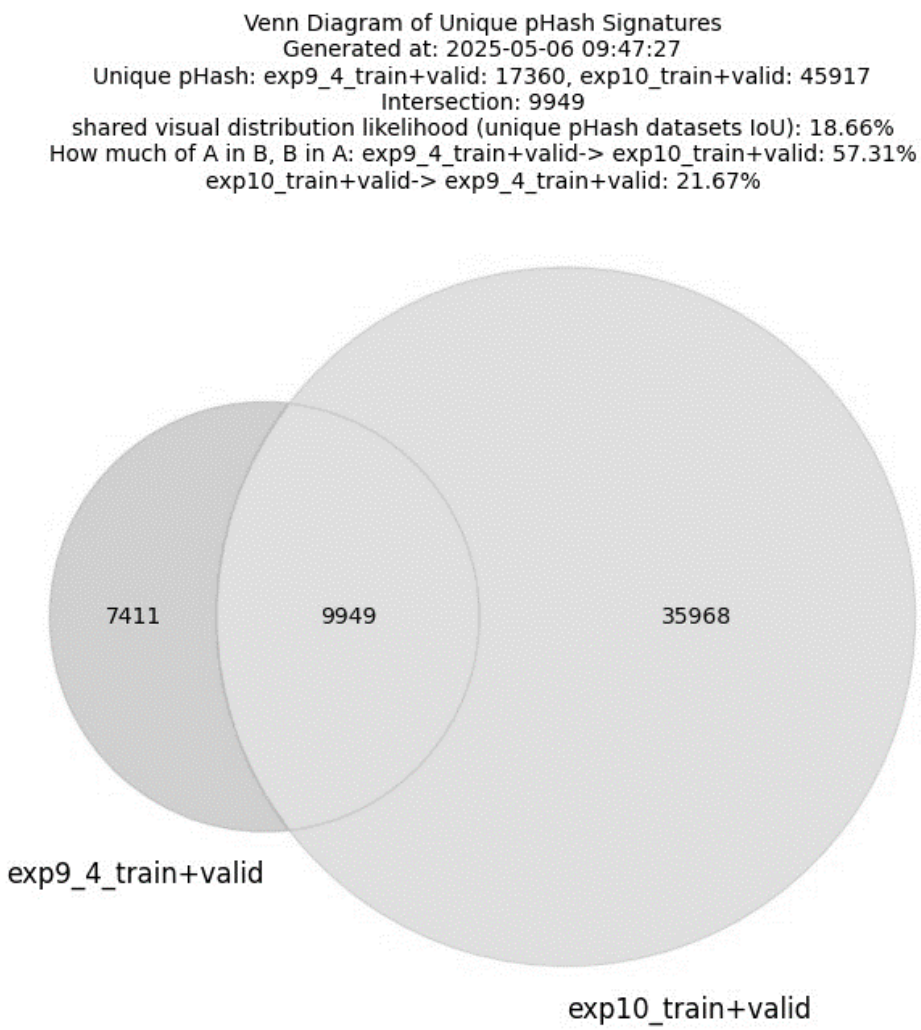


Figure **Error! No text of specified style in document.**.43 Similarity test between training syllabus 9.4 and 10

Despite the training syllabus in exp.9.4 yielding a smaller set, it still enabled the training of a model that we can trust more in terms of its ability to handle Black Swan operations compared to the training syllabus that only focused on OOC training with some in-context images.

What is truly important to ensure performance during Black Swan scenario operations is not the size of the dataset, but rather its awareness of context, particularly how it relates to Black Swan events. However, that's the real challenge from the perspective of safety assurance and constructing a meaningful safety case. Because the process needs to expose unmitigated, OOD datasets, we can demonstrate that “unmitigated, OOD” data shift had been identified and deliberately mitigated by the training and validation datasets pictorially sense. Looking at the entire experimentation process, it appears that a systematic incremental approach where we systematically introduce Black Swans sets, test that they are OOD, then check performance against them, demonstrates a clear intended design for Black Swan ops.

We propose that a safety assurance process needs to identify a Black Swan and then validate that it is indeed a Black Swan in relation to the training and validation processes. Hence, the process in section 5.1.7 was proposed.

In short, we can summarise that:

1. **Hypothesis Validation**: Models trained without Black Swan data (Group 1) failed catastrophically on Black Swan tests (24–35% mAP@50). Explicit inclusion (Group 2) boosted performance to **67%**.
2. **Dataset Similarity**: The Black Swan test sets showed 0% IoU with the training data in Groups 1 and 2, confirming their OOD nature.
3. **Statistical Significance**: Group 4's performance was systematically higher, supporting the necessity of Black Swan coverage to assure model performance over Black Swan scenario.
4. **Black Swan-driven Systematic Development Pipeline:** A bonus realisation that we unexpectedly learned from conducting these experiments is that we were essentially performing a systematic and incremental approach that can demonstrate a deliberate design against failures under Black Swan operations. This approach can be used as an assurance process for demonstrating compliance in safety cases. We integrated this idea into our overall process in stage 6 (see also section 4.9).

For key findings and conclusions, refer to Chapter 9, Section 9.5.2.

H.11 Black Swan Development and Validation experiments implementation[12]

AMLAS mentions a distinct type of requirement:

- Performance
- Robustness

The primary difference in performance relates to prediction accuracy and precision. At the same time, robustness is associated with various real-world scenarios over which the model can generalise well. The general expectation is that training a model on training data and testing it on out-of-statistical distribution (relative to the training dataset) yields lower performance than testing the same model within the training context. Thales had published a work confirming our expectation, referring to such datasets as either “InD” or “OOD”. [6]. The authors asserted that:

> "Mathematically, when an ML-based model is successfully trained on a given dataset, then the model is expected to produce accurate predictions for unseen, InD test data. Conversely, the accuracy of the model is expected to fall when processing data instances drawn from OOD test data”.

AMLAS guidelines suggest that models test datasets that are required to be different but “similar” for the sake of robustness assurance.

How do we validate that a test dataset is different but similar?

Hence, we need to incorporate a similarity test for a meaningful safety-related purpose Validating a dataset means comparing the generated dataset against the specified cuneiforms. The output of stage 5 is a valuable minimum variety of datasets that can be used to train and test a model. In this stage, we would validate that the generated datasets correctly capture their characteristic CuneiForm classes. However, we will reserve the CuneiForm validation process for the AVOIDDS case study and focus on training a model and comparing it to the performance when a model lacks Black Swan training. This section will describe three limited proofs of concept for developing and validating performance over Black Swan scenarios. In this stage, we will investigate the impact of discovering Black Swans and how we can technically understand them from a dataset point of view.

H.11.1 Our Approach Using imagehash (pHash):

In contrast to ADD, we take a perceptual, vision-based route. We compute a perceptual hash (pHash) for each image using the imagehash library. Each image is converted to a compressed signature that captures its overall visual structure. We then compare these hash values across datasets:

[12] See also section 7.2

- If many images in the test have hash values that match those from training (or vice versa), we consider them “InD” and similar.
- Hash-based visual matching is coarse, meaning if a dataset indicates the presence of a high percentage of similar hashes (in the same dataset), this does not mean all hashes represent the same image. It only means visually the same, but not exactly the same; some may be, but the likelihood is considered negligible.
- We will not use it to determine duplication; we will only use it to estimate the likelihood of visual similarity.
- We will use it only to compare across the same experiment dataset for similarity.

Because our method maps an image directly to a perceptual signature, it bypasses the need to compute detailed activation histograms and anomaly scores. Thus, the visual similarity is less rigorous than the ADD method, and this is sufficient for the purpose of this PhD. For example, suppose we see a high percentage (say 97–99%) of matching hashes. In that case, we know that the test dataset is very similar to the training set regarding visual content, but not necessarily exact copies.

What can one hash be given to multiple images? Let’s take an example:

Table H.51 Example perceptual hash given to multiple visually similar images

Dataset	**Given perceptual hash**
exp4_testing	f2c0e0dec6a4b6a6
exp4_testing_img_1019.jpg; exp4_testing_img_1021.jpg; exp4_testing_img_1045.jpg; exp4_testing_img_1079.jpg; exp4_testing_img_1086.jpg; exp4_testing_img_1094.jpg; exp4_testing_img_1095.jpg; exp4_testing_img_1096.jpg; exp4_testing_img_1101.jpg; exp4_testing_img_1105.jpg; exp4_testing_img_1107.jpg; exp4_testing_img_1108.jpg; exp4_testing_img_1158.jpg; exp4_testing_img_140.jpg; exp4_testing_img_235.jpg; exp4_testing_img_333.jpg; exp4_testing_img_357.jpg; exp4_testing_img_4.jpg; exp4_testing_img_412.jpg; exp4_testing_img_414.jpg; exp4_testing_img_442.jpg; exp4_testing_img_468.jpg; exp4_testing_img_564.jpg; exp4_testing_img_63.jpg; exp4_testing_img_687.jpg; exp4_testing_img_699.jpg; exp4_testing_img_816.jpg	

To define what we mean by those terms:

A Training dataset X is considered InD or in the same distribution as Training dataset Y when drawn from the same original scenarios of dataset Y, yet somewhat different in instances. This means that dataset Y data's features, style, and statistical properties are similar to dataset X's (not exact copies), and that both datasets are driven from the same perspective about real-world scenarios.

So what?

Well of the training dataset includes images of drones flying over grass fields, the architect is assuming that the operational domain will involve flying over open grass fields. Suppose the datasets include a particular person from a particular part of the world carrying a drone in his or her hands. In that case, the architect is assuming a requirement for detecting a drone specifically held by a person from a certain demographic. With such perspective in mind, it is legitimate to question the architect who would choose to include random scenarios (OOC) on whether the customer or the operational domain expects such scenarios or not.

By “perspective”, we mean a general idea of a scenario, like flying over an open field or a Cuneiform abstraction. A CuneiForm represents a general perspective about a scenario. We consider Imagehash similarity indicates that a group of images come from a similar perspective or idea about some potential scenario. This means, if we group a set of images by a single pHash, we consider that set of images as various instantiations of some unique perspective or idea of a potential scenario.

If datasets are different (0 pHashes), then they come from different perspectives on scenarios. For example, dataset X is made up of random images of drones, and the testing dataset is also made up of random images of drones, but in different situations. In our approach using image hash, we assumed some concrete classification to help us be more specific about the nature of the distribution (see table E.46).

H.11.2 Definitions of Key Performance Metrics in Object Detection

Precision: Precision measures the proportion of true positive detections out of all detections made by the model. It indicates how many of the detected objects were correct. The following is the general formula for precision.

$$Precision = \frac{Correctly\ Predicted\ Positives}{All\ Predictied\ Positives}$$

Example: A precision of 95.0% means that 95 out of every 100 detected drones were correctly identified, while 5 were incorrect detections (false positives). The experiment observed false positives in 53% of Black Swan cases, indicating that the model sometimes misidentifies objects in unfamiliar scenarios.

Recall: Recall measures the proportion of actual objects the model correctly detected. It represents the model's ability to identify all relevant objects. The following is the general formula for recall:

$$Recall = \frac{Correctly\ Predicted\ Positives}{All\ actual\ positive, presence\ of\ ToIs\ (detected\ and\ undetected)}$$

Example: A recall of 95.3% means that out of 100 actual drones present, the model successfully detected 95, while it missed 5 drones (false negatives). In the experiment, the model failed to detect 82% of Black Swan cases, indicating that recall is lower in complicated scenarios.

Mean Average Precision (mAP): is a commonly used metric in object detection that evaluates the precision-recall trade-off across different confidence levels. It is calculated by computing each class's Average Precision (AP) and then averaging across all classes.

Example: If an object detection model achieves a high mAP, it correctly detects objects while minimising false positives and negatives. In the given experiment, an mAP of 96.7% indicates that the model performs well in standard detection scenarios but may not generalise well to Black Swan situations.

- As you lower the threshold (be more lenient about saying “positive”), recall goes up (you catch more of the good apples), but precision often goes down (you also start calling some bad ones “Good”).
- **Average Precision (AP)** is the **area under that Precision–Recall curve**. In practice, we compute it by summing (precision at various recall levels) × (small recall step), or by taking the precision at each rank in a sorted list of scores.

- If you have multiple categories or multiple queries (e.g., detecting apples vs. oranges vs. bananas) or multiple IoU thresholds in an object-detection task, you typically compute AP for each class/threshold and then average them → that's **mean AP (mAP)**.

Key takeaway: AP/mAP focuses on how many TOIs are truly present **and** how many of your "predictions" are correct. It tends to penalise models harshly if models cry out TOIs, but many of them turn out not to be TOIs. In extremely skewed settings (e.g., only 10 images have TOIs and 990 images have no TOIs), PR and AP give you a clearer picture of how many false alarms you'll have when trying to catch most good apples.

False Positive Rate FPR: The ratio between incorrectly detected positives with respect to all actual negatives.

$$FPR = \frac{Incorrectly\ Predicted\ Positives\ (negatives\ that\ classed\ as\ positives)}{All\ actual\ negatives}$$

True Positive Rate TPR: the ratio between the number of correctly predicted positives over all actual positives.

$$FPR = \frac{Correctly\ Predicted\ Positives\ (negatives\ that\ classed\ as\ positives)}{All\ actual\ negatives}$$

Area Under the Receiver Operating Characteristic (ROC): is a single number ∈ [0, 1] that tells you, on average, how well your model ranks a randomly chosen predicted positives higher than a randomly chosen predicted negatives.

Key takeaway: ROC cares about TPR vs. FPR, regardless of how many positives vs. negatives you have. It asks, "If I pick one good apple and one bad apple at random, how often does the model give the good apple a higher score?" That probability is exactly AUROC.

- **AUROC** asks: "If I randomly pick one good and one rotten apple, how often does the model score the good one higher?" Even in heavily imbalanced cases, AUROC remains relatively stable (because it measures ranking across all pairs).
- **AP / mAP** asks: "Looking at the top-K predictions (or at each recall step), how many of them are truly good?" It penalizes you harshly if you let a lot of rotten apples into the high-confidence region.
- Use **ROC/AUROC** when you need a threshold-independent measure of separation, and the problem domain is expected to have TOIs the same as the number of frames where there are no TOIs. However, if we have a problem where there are hardly any TOIs or there are plenty of frames or expected frames with TOIs, then ROC can mask the poor real-

world impact of those extremes, especially if we have relatively small false positives compared to the number of frames that have no TOIs in them.

- Use **Precision–Recall / AP / mAP** when actual positives or negatives are rare (e.g., fraud detection, medical diagnosis, object detection in images).

Confidence Threshold: The confidence threshold is the minimum probability score required for a detected object to be valid. The detection is ignored if the model assigns a probability lower than 50%.

Example: If the confidence threshold is set at 50%, the model will only accept detections where it is at least 50% confident that an object is present. A higher threshold (e.g., 80%) would reduce false positives, while a lower threshold (e.g., 30%) might increase recall but reduce precision.

Overlap Threshold (IoU Threshold): The overlap threshold, often measured as Intersection over Union (IoU)**,** determines the minimum bounding box overlap required for a prediction to be considered correct. It measures how much the predicted bounding box overlaps with the ground-truth bounding box. The following is the general formula for IoU:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Example: If the IoU threshold is set at 50%, a detected drone must overlap with the actual drone's bounding box by at least 50% to be counted as a true positive. If the threshold were higher (e.g., 75%), the model would require a more precise detection to be considered correct.

Training dataset: The dataset split is used to train the model.

Testing dataset: The dataset is used with the training process to update model parameters during the learning process.

Validation dataset: the final dataset used to make the final verdict on the ML model performance.

Context of training, validation, and testing: The context of an image or a dataset is characterised by the CuneiForms developed by the architect to capture the problem domain of concern. The architect's definition of the intended context and whether a set of images is outside or within that context is subjective.

In-context/OOC: referring to a dataset within the context of the specific problem domain or outside of it. For example, let's say the training dataset is “in-context”. We mean that the images

in the training dataset fall within the context of the concerning problem domain, like drones flying over London King's Cross train station, on a particular day. “OOC” we mean, anything outside the context of the problem domain, for example, a drone flying over the desert or displayed in some store.

InD/OOD: An image X is said to be InD with image Y when image X has similar visual characteristics to Y, but not necessarily a copy of Y. We take a slightly different approach to describe the distribution from the work done by Thales researchers, called ADD (Anomaly-based Dataset Dissimilarity) [6]. The researcher approached dissimilarity from a mathematical standpoint (statistical characteristics of images). We looked at the distribution problem from a different standpoint, and the following is a description of how we considered it:

H.11.2.1 Why pHash and not MD5 for computing Datasets IoU?

Both approaches compute a fixed-length string that "summarises" the content of an image, but they do so in fundamentally different ways and with very different goals. The following is a simple comparison to explain how the standard MD5 function[13] (as implemented in get_md5_hash) differs from an image perceptual hash method provided by the imagehash library[14] (such as pHash):

- **Purpose and Sensitivity:**
  - MD5 is **exact**: It is sensitive to every single bit change. It is ideal for detecting exact duplicates.
  - Image perceptual hash (pHash) is designed to capture the **visual essence** of an image. It tolerates small changes (like compression artefacts, slight rotations, or scaling) and gives similar results for visually similar images.
- **Data Processing:**
  - MD5 works directly on the binary file without understanding image content.
  - pHash or similar methods process the image content (converting to grayscale, resizing, transforming frequency components) so that the hash reflects human-perceived features rather than raw data.
- **Applications:**
  - MD5 hashing to verify that two images are identical, such as for integrity checks.

[13] Refer to the following link on geeksforgeeks.com that describes the MD5 hash in Python method.
[14] Visit the following link that describes the technical details of imagehash methods (pHash).

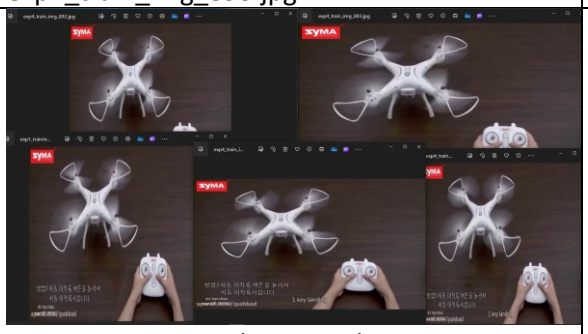
- A perceptual hash is more useful when the goal is to detect visually similar images, even if the files are not the same (for example, when images may have been re-encoded or subtly altered).

In our particular case, we face a problem of complexity, where we used multiple sources of datasets and that made it harder for us to determine what differences we have intentionally or unintentional introduced in the datasets. Therefore, we needed to validate how similar datasets are to each other. The comparison of the similarity between related experiments presents a highly nuanced challenge. When we consider perceptual hashing (pHash), it encapsulates the precise similarities that MD5 would reveal, and other visual similarities present within extensive datasets. Consequently, we believe that phash is more robust in accommodating the variations of images within datasets, thereby providing a superior understanding of the visual resemblance between the two datasets.

Table H.47 demonstrates how the similarity method works and how we used to compare similarity between experiments. We took two datasets, exp.1 and 4 for example, then computed perceptual hashes for each image and then searched for similar hashes in the other.

Table H.52 Example of how images in different datasets can have the same pHash

File in exp1_training	File in exp4_training	Signature
exp1_training_img_1018.jpg	exp4_train_img_882.jpg	804d793269d63e6e
	exp4_train_img_883.jpg	804d793269d63e6e
	exp4_train_img_889.jpg	804d793269d63e6e
	exp4_train_img_892.jpg	804d793269d63e6e
	exp4_train_img_893.jpg	804d793269d63e6e
exp1_training_img_1320.jpg	exp4_train_img_882.jpg	804d793269d63e6e
	exp4_train_img_883.jpg	804d793269d63e6e
	exp4_train_img_889.jpg	804d793269d63e6e
	exp4_train_img_892.jpg	804d793269d63e6e
	exp4_train_img_893.jpg	804d793269d63e6e
exp1_training_img_562.jpg	exp4_train_img_882.jpg	804d793269d63e6e
	exp4_train_img_883.jpg	804d793269d63e6e
	exp4_train_img_889.jpg	804d793269d63e6e

	exp4_train_img_892.jpg	804d793269d63e6e
	exp4_train_img_893.jpg	804d793269d63e6e
	 Training images in the Exp.4 dataset	All example images in both datasets share the same pHash: 804d793269d63e6e Such images are classed as OOC, InD

We then output a Venn diagram showing how two datasets are identical. Figure H.40 demonstrate the Venn diagram which captures how the exp.1 and 4 training and validation datasets are similar with a shared perceptual hash in amber and unique hashes in not-amber colour.

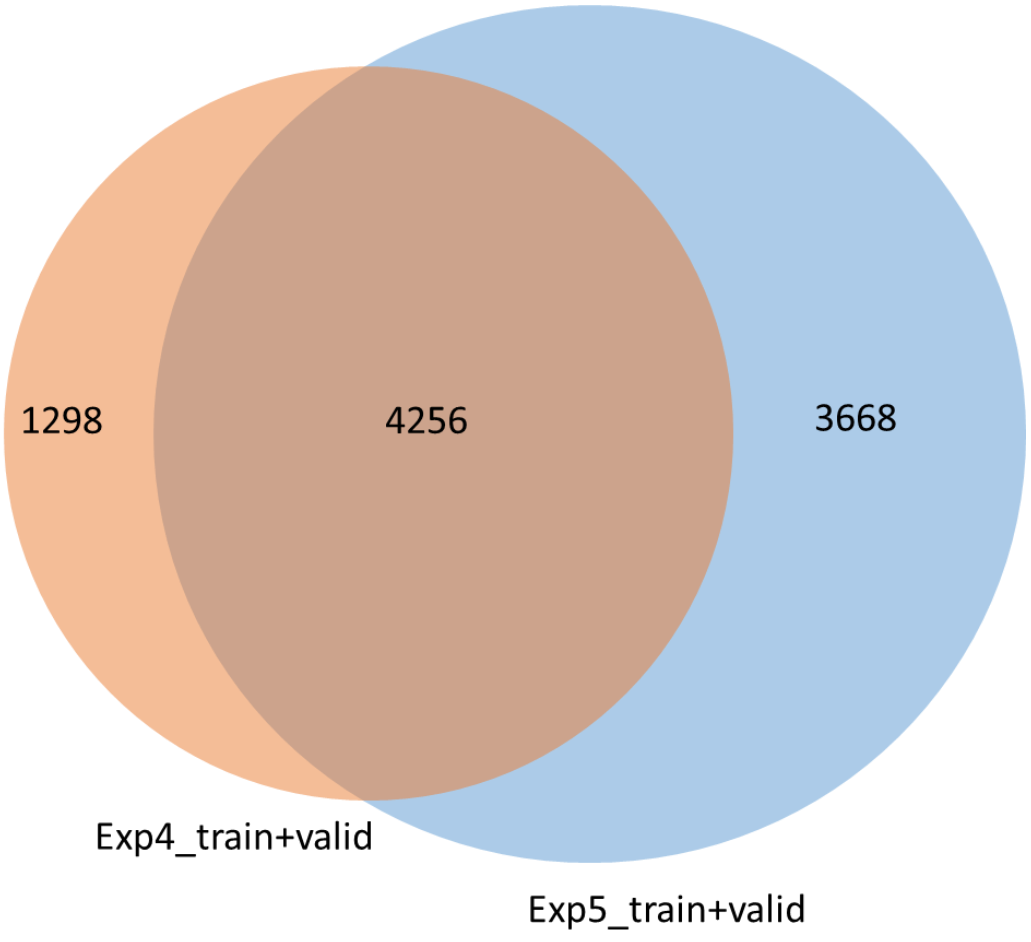


Figure H.44 Similarity test between exp.1 training and validation dataset and 4 training and validation

H.11.2.2 In/OOC and datasets distribution types

- A 0% matching hash would indicate that the dataset is truly “OOD” and completely visually different (generated from different requirements, fundamentally different CuneiForms).
    - If a dataset is in-context, then it is a true in-context, Black Swans relative to the reference dataset.
    - If a dataset is OOC, then it is a true OOC, Black Swans relative to the reference dataset.
- If the comparison measure between datasets falls in the range of 0>matching hashes <=20% of the comparable datasets, then they are acceptably “InD” but still visually different (generated from different requirements, fundamentally different CuneiForms).
    - If a compared dataset is in-context to the problem, then it is an in-context, potential-Black Swan relative to the reference dataset.
    - If a compared dataset is OOC to the problem, then it a OOC, semi-Black Swans relative to the reference dataset.
- If compared datasets have >20% of shared hashes, they are acceptably “InD” and visually similar datasets (generated from different requirements, fundamentally the same CuneiForms).
    - If a compared dataset is in-context to the problem, then it is an in-context, normal dataset to the reference dataset.
    - If a compared dataset is in-context to the problem, then it is an in-context, normal dataset to the reference dataset.

H.11.2.3 Typical operations

Referring to situations that the architect considers relatively typical day-to-day scenarios or the most likely scenarios a robot may encounter. For example, assume the most common drone on the market is the DJI Phantom; therefore, an image of a DJI drone would represent a typical operational situation a robot may encounter. However, a fully camouflaged DJI drone with the exact patterns of the operational domain would represent a non-typical, Black Swan-like situation. The architect assumes that such extreme designs are unlikely, thus non-typical.

H.11.2.4 Determining Dataset Distribution Overlap (Datasets Visual IoU)

We needed a quantitative index that could tell us holistically how one dataset is similar to another to validate the experiments' results. The method must help us to determine whether one dataset is InD with respect to another is to using a set-similarity metric applied to their unique representations. The indicative index will help us to determine:

1. Whether dataset A is in-visual-distribution with dataset B (index >0).
2. If a model trained on dataset A and dataset B is completely out of visual distribution (index = 0) , then it is likely that they are Black Swans to each other, meaning as a whole they belong to completely distinct and different requirements.

In our experiments, we leverage the unique image signatures (represented as pHash hashes) to compute the **Jaccard index** [7] (also known as intersection over union (IoU)) between two datasets. For the purpose of dataset visual similarity, we will refer to it as: Jaccard's dataset visual similarity likelihood since it indicates a reasonable rough likelihood estimate of how two datasets are similar or different to each other.

For example, consider the following calculations from our experiments:

- Unique pHash in exp1_training = 4489
- Unique pHash in exp2_training = 4612
- Intersection of unique pHash (common to both datasets) = 4388

The Jaccard similarity coefficient, **J(A,B)**, is given by:

$$J(A,B) = (|\text{Intersection}|) / (|\text{Unique pHash count A}| + |\text{Unique pHash count B}| - |\text{Intersection}|)$$

Intersection means common pHash values between A and B. Substituting our numbers:

$$J \approx 4388 / (4489 + 4612 - 4388) = 4388 / 4713 \approx 0.93 (93\%)$$

This high value and the "no-dup coverage" percentages (97.75% for exp1_training → exp2_training and 95.14% for the reverse) indicate that nearly all of the unique images in one dataset are present in the other. In other words, there is a high degree of overlap, suggesting that the datasets are nearly InD.

Using the Jaccard coefficient is effective because it captures the diversity of each dataset (via the unique pHash counts) and the actual overlap (via the intersection) within a single scalar value ranging from 0 to 1. Although alternative metrics, such as the F1 score between unique sets or embedding-based similarity metrics, could be used, the Jaccard index is conceptually straightforward and quantitatively robust for comparing image datasets based on unique identifiers.

Thus, within our experimental framework, calculating the Jaccard similarity coefficient (or equivalent Coverage metrics) is the preferred quantitative method to accurately assess the degree to which Dataset A is in visual distribution about Dataset B. This InD measure offers

critical evidence supporting the effectiveness and thoroughness of our dataset curation for safety-critical ML applications. Below is a table that describes the types of dataset similarity:

Table H.53 Dataset Comparison Matrix Based on Matching Hash Percentage and Problem Domain Context

Dataset Similarity IoU %	In the Context of the Problem Domain	OOC of the Problem Domain
0%	**Fully OOD, In-Context of CuneiForms** Definition: The dataset is completely visually different from the reference set, despite being contextually intended for the application (through mapping to a CuneiForm). This category can be considered true Black Swans if the images belong to Black Swan Cuneiforms. If, however, the photos belong to typical operations, this means one of the datasets is Black Swans compared to the other (if we accept the typical operations as the no **Conformance to CuneiForm:** The dataset is part of the problem domain defined in the systems approach. In other words, there is a CuneiForm specified for an image to be an instantiation of.	**Fully OOD, OOC of CuneiForms** Definition: The dataset is entirely different (no shared hash) from the reference; it may be generated from an entirely different requirement. **Non-Conformance to CuneiForm:** The dataset is not part of the problem domain defined in the systems approach. In other words, no CuneiForm are specified for an image to be an instantiation of.
	Example: A set of images, which are meant for the same problem domain, that bear no visual similarity (zero matching hashes) to the training set.	Example: Images from a completely different domain (e.g. aerial photographs) compared to a handwritten-digit reference.
	Expected: Relative rare occurrence; a high divergence that suggests true anomalies.	Expected: Also rare; indicates a dataset that does not belong to the application domain at all.
0%> to ≤5%	**Potential OOD, In-Context of CuneiForms** Definition: There is some overlap (up to 20% matching hashes), so the dataset is acceptably OOD, and only a small portion is visually the same. These cases might be "unexpected" examples within the same domain. **Conformance to CuneiForm:** The dataset is part of the problem domain defined in the systems approach. In other words, there is a CuneiForm specified for an image to be an instantiation of.	**Potential OOD, OOC of CuneiForms** Definition: The dataset shares a small fraction (≤20%) of its visual signature with the reference, indicating that while some elements overlap, many images are distinct. **Non-Conformance to CuneiForm:** The dataset is not part of the problem domain defined in the systems approach. In other words, no CuneiForm are specified for an image to be an instantiation of.
	• Example: A largely in-domain image set that contains unusual samples (e.g. heavily rotated or occluded images) that lower the overall hash matching rate.	• Example: A dataset generated under different requirements, where only a few images share visual characteristics with the reference.

5%> to >90%	**Potential InD, In-Context of CuneiForms** Definition: Normal, meaning the compared datasets are visually normal to each other. A high proportion of shared hashes (more than 20%) suggests that the dataset is largely visually similar to the reference, as expected for in-domain samples. **Conformance to CuneiForm:** The dataset is part of the problem domain defined in the systems approach. In other words, there is a CuneiForm specified for an image to be an instantiation of.	**Potential InD, OOC of CuneiForms** Definition: Normal, meaning the compared datasets are visually normal. Even if the dataset is generated under different requirements, more than 20% of shared hashes indicate a high degree of visual similarity (perhaps sharing a common structure, style, or form). However, one of them is in the context of the problem domain, and the other is out of context. **Non-Conformance to CuneiForm:** The dataset is not part of the problem domain defined in the systems approach. In other words, no CuneiForm are specified for an image to be an instantiation of.
90%≤ to ≤100%	Example: A test set of images for a handwritten digit classifier that largely matches the training distribution. The dataset is part of the problem domain defined in the systems approach. **Fully InD, In-Context of CuneiForms** Definition: nearly visually identical datasets which prompt certain expectations of performance. For example, in theory, a model should perform similarly across two visually similar datasets. Also, if the training, validation and testing datasets are identical, the model may not generalise well on unseen data, let alone Black Swans. **Conformance to CuneiForm:** The dataset is part of the problem domain defined in the systems approach. In other words, there is a CuneiForm specified for an image to be an instantiation of.	Example: Two datasets that come from different problem domains, concerning the same TOI, sources but are based on the same fundamental characteristics (e.g. similar imaging modalities) can still have high matching percentages. The dataset is not part of the problem domain defined in the systems approach. **Fully InD, In-Context of CuneiForms** Definition: the same as the other type of twin. **Non-Conformance to CuneiForm:** The dataset is not part of the problem domain defined in the systems approach. In other words, no CuneiForm are specified for an image to be an instantiation of.

Note: We use Venn diagrams to visually capture the distinctions among various dataset configurations (IoU). This process may be employed to analyse the evolution of the dataset over time following its deployment. However, this process may necessitate supplementary safety checks if the dataset deviates from its original situation at any point during deployment. For instance, testing the newly trained model against the previous dataset distribution is advisable.

From the above table, we redefine what True Black swans are and what they are not. We needed this definition because when we applied stage 6, we came across a situation where a

- **Black Swan**: 0% pHash overlap **and** the scenario is operationally surprising (e.g. a drone you've never even imagined). It represents an operational data shift.
- **Typical operations novelty**: 0% pHash overlap **but** the scene is entirely "business as usual" from the model's point of view (e.g. just a new colour scheme, slightly different lighting, a camo pattern that changes low-level pixels but nothing else).

H.11.3 ML training and testing strategies experiments

The objective of the ML development experiments is to investigate the impact of context-aware dataset construction on the robustness and effectiveness of object detection models when faced with typical and non-typical operational scenarios, including potential Black Swan events. We wanted to investigate how the output of stage 4 (having gone through stages 1, 2,3) can help the architect to ensure and able to assure (presenting compelling argumentation) that the output model is trustworthy, and risks have been maintained ALARP.

This subsection describes the experimental process employed to test the hypothesis that in-context, non-typical (Black Swan) images enhance the quality of the machine learning (ML) model. Furthermore, it aims to evaluate whether performance testing over Black Swan scenarios provides more informative assurance argumentation and assessments of ML quality than testing only over typical scenarios. The following outlines the key research questions, experimental setup, performance metrics, and comparative analyses required to address these goals. So, our hypotheses are the following:

Hypothesis 1 (Hyp1): A model trained and validated primarily on typical operations or OOC images may perform well on similar "typical" validation/test data. Still, it may fail dramatically on "black swan" (non-typical scenarios, fully OOD) images unless those black swan scenarios are explicitly represented in training. In other words:

- **Black Swan Context aware dataset:** Including in-context black swan examples in the training set is essential for better performance and more convincing assurance arguments on black swan test data.

- Furthermore, testing over a dedicated Black Swan scenarios test set is a useful metric to ensure that the trained model is somewhat ready to handle rare and high-impact scenarios.

Specifically, we ask:

Q1: Does a model trained on only typical or OOC images perform better or poorly on black-swan scenarios, and, conversely, does explicitly including black-swan images in training measurably improve the model's performance on black-swan test sets?

These questions underpin the motivation to systematically compare the performance of 9 different training-validation-testing strategies under two core experiment types:

- Testing over Black Swan robustness evaluation:
    - **Expected results:** if trained on typical and OOC, the model may experience noticeable performance reduction over Black Swan test datasets, unless trained on similar Black Swans.
- Testing over typical in-context operational performance evaluation:
    - **Expected results:** A misleading measure of good performance, as they may not perform as intended in Black Swan events.

H.11.3.1 Methodology

All models were developed using the **Roboflow 3.0 Object Detection (Fast)** architecture, with **COCO-trained weights (COCOn)** as the initial checkpoint for some of them; we used an earlier version of the models. No image augmentations were applied, ensuring that observed performance differences arise solely from the dataset compositions. The main online datasets we used for OOC random images are the following:

- DRONES_NEW Computer Vision Project
- droneandbird Computer Vision Project
- Drones Computer Vision Project
- Drone Project Computer Vision Project
- Drone Detection Computer Vision Project

It is worth noting that all random, OOC images used across the experiments are sourced from the above datasets. Any image that does not conform to a CuneiForm is considered part of the random images set; however, it has a background that somewhat resembles the operational domain.

H.11.3.2 Experiment setup:

Below is a concise overview of Experiments 1 through 9.3 and their interrelationships. Each experiment varies primarily by (1) the composition of training/validation/test splits (e.g., the proportion of "in-context" vs. "OOC" vs. "non-typical black swan" images), (2) whether and how CuneiForm scenarios are included, and (3) which pre-processing, or augmentation steps were

applied. These differences help probe the model's robustness under various real or rare (black swan) operational conditions.

There are two main assumptions to make:

- **Assumption 1:** The output models will be deployed with an IoU threshold of exactly 0.5; thus, mAP@50 is the key metric.

We want to emphasise an aspect regarding the comprehensiveness of the experiments conducted. A comprehensive experimentation process requires a substantial number of datasets and resources. We acknowledge that this research is not focused on producing a new machine learning architecture; instead, it centres around the architectural process aimed at addressing assurance in safety-critical problems. In light of this perspective, we believe that the experiments and results obtained are adequate to illustrate potential outcomes; although not entirely conclusive, they are sufficient to suggest a possible direction for future inquiries. based on this, we consider the following assumption:

- **Assumption 2:** The dataset size chosen is sufficient for all models. So we claim that the size for a given task is sufficient.

Table H.54 typical operations CuneiForms

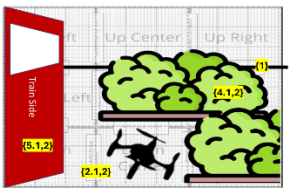
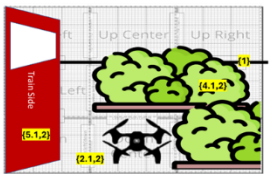
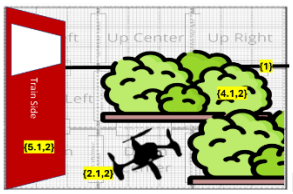
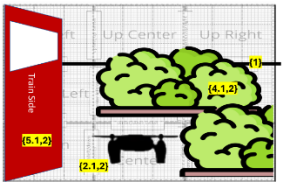
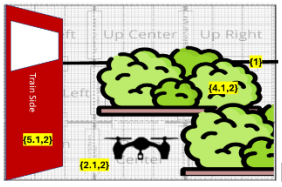
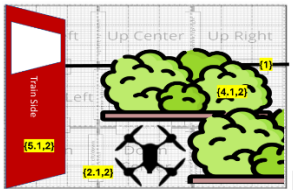
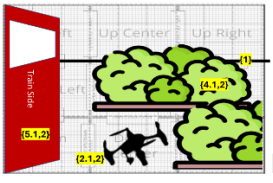
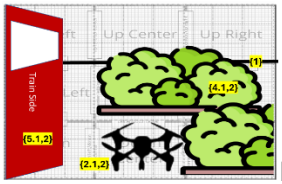
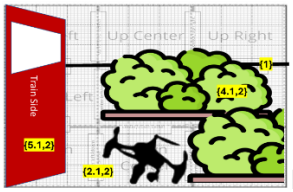
Training Scenarios	Validation Scenarios	Testing Scenarios
 H.42	 H.45	 H.48
 H.43	 H.46	 H.49
 H.44	 H.47	 H.50

Table H.55 Black Swan Scenarios Batch A and B CueniForms

In-context, Black Swans CuneiForm (Batch B)	In-context, Black Swans (Batch A)

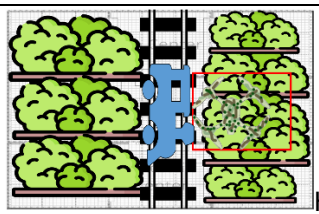
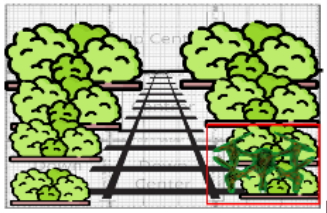
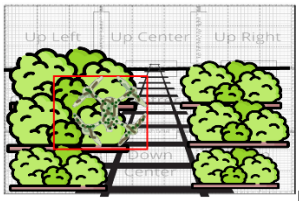
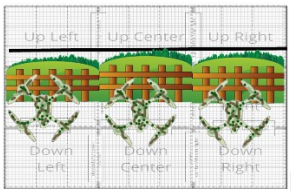
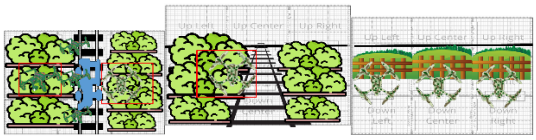
 H.51	 H.37
 H.52	 H.38
 H.53	 H.36

Table H.56 Test sets of Exp.8.2 and Exp.3 comparison between

Black Swans CuneiForm (Batch A)	**Black Swans CuneiForm (Batch B)**
	
TOIS are located at the following positions: centre right, centre, and centre left.	TOIs are located at the following positions: down right
TOIs 3D attitudes: Top, Front down left 	**TOIs 3D attitudes:** Front down center, Front down right 

Below is a table that captures each experiment strategy:

Table H.57 ML development experiments definition

Experiment	**Training Strategy**		**Testing strategy**
	Training Dataset	**Validation Dataset**	**Testing Dataset**
Experiment 1: see link	**Total:** 6753(71%)	**Total:** 1679 (18%)	**Total:** 1063 (11%)
	Random, OOC: 6753 (100%)	**Random, OOC**: 1679 (100%)	**Random, OOC**: 1063 (100%)
	No CuneiForms	No CuneiForms	No CuneiForms

Total: 9495 images	**Note:** The validation and test sets used in this experiment are identical to those in Experiment 2. The random images in the training set are identical to those in the training set used in Experiment 2. No pre-processing steps were applied. No augmentations were applied.		
Experiment 2: see link **Total:** 9495 images	**Total:** 6753 (71%)	**Total:** 1679 (18%)	**Total:** 1063(11%)
	In-context, Black Swans: 224 **Random, OOC:** 6529	**Random, OOC**: 1679 (100%)	**Random, OOC**: 1063 (100%)
	Black Swans A CuneiForm: H.36	No CuneiForms	No CuneiForms
	Note: The validation and test sets are the same as those in Experiment 1. The random images in the training set are identical to those in Experiment 1's training set. No pre-processing steps were applied. No augmentations were applied.		
Experiment 3: See link **Total:** 10658 images	**Total:** 8000 (75%)	**Total:** 1500 (14%)	**Total:** 1158 (11%)
	In-context non-typical black swan images: 2390 (30%) **OOC images:** 5610 (70%)	**In-context non-typical black swan images:** 701 (47%) **OOC situations:** 799 (63%)	**In-context, non-typical, Black Swan operational images:** 1158 (100%)
	Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41		
	Note: The test dataset is identical to the one used in Experiment 4. The instantiations of the cuneiforms are different in each dataset. No pre-processing steps were applied. No augmentations were applied.		
Experiment 4: see link **Total:** 10658 images	**Total:** 8000 (75%)	**Total:** 1500(14%)	**Total:** 1158 (11%)
	In-context images: 0 **OOC images:** 8000 (100%)	**In-context images:** 0 **OOC images:** 1500 (100%)	**In-context, non-typical, Black Swan operational images:** 1158 (100%)
	No CuneiForms	No CuneiForms	**CuneiForm Scenarios:** H.36, 37, 38, 39, 40, 41

<table>
<tr><td></td><td colspan="3">Note: The test dataset is identical to the one used in Experiment 3. No pre-processing steps were applied. No augmentations were applied. The training dataset of Exp.4 has a different set of OOC images from the one we used in Exp.1.</td></tr>
<tr><td rowspan="4">Experiment 5:

See link

Total: 14384 images</td><td>Total: 9924(69%)</td><td>Total: 1399(10%)</td><td>Total: 3061(21%)</td></tr>
<tr><td>In-context images: 0

OOC images:

9924 (100%)</td><td>In-context images: 0

OOC images:

1399 (100%)</td><td>In-context typical operations images:

3061 (100%)</td></tr>
<tr><td>No CuneiForms</td><td>No CuneiForms</td><td>CuneiForm scenarios: H.48, 49,50</td></tr>
<tr><td colspan="3">Note: The testing dataset is identical to experiment 6. Applied Pre-processing: Resize: Stretch to 320x320. No augmentations were applied. No augmentations were applied.</td></tr>
<tr><td rowspan="4">Experiment 10:

See link

Total: 101548 images</td><td>Total Training: 85677 (84%)</td><td>Total Valid: 14713 (14%)</td><td>Total Test: 1158 (1%)</td></tr>
<tr><td>In-context typical operations images:

3270 (4%)

OOC situations:

82257 (96%)</td><td>In-context typical operations images:

1202 (8%)

OOC situations:

13511 (15%)</td><td>In-context, non-typical, Black Swan operational images:

1158 (100%)</td></tr>
<tr><td>CuneiForms scenarios: H.42, 43,44</td><td>CuneiForms scenarios: H.45, 46,47</td><td>Black Swans Batch A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td></tr>
<tr><td colspan="3">Note: this is an extended version of experiment 7. We added a significant number of random, OOC images to the test dataset to assess how increasing the number of images would affect performance compared to Black Swans.

Pre-processing:</td></tr>
</table>

	• Resize: Stretch to 320x320

H.11.3.3 Testing approach

Across Experiments Dataset Visual Similarity Test: We check how datasets compare in terms of similarity using either image hashing methods or visual techniques to confirm how close experimental datasets are to one another. This activity was necessary to consolidate the assumptions and ensure the expected results are comparable for a given ML architecture. Although these methods have advantages and disadvantages, we consider them sufficient indicators of how similar datasets are.

Across the same experiment, Visual Similarity Test: We use the library "ImageShare" in Python to determine whether two datasets (testing and validation+training) are somewhat the same. We view the method as an indicative likelihood of similarity between datasets. If the similarity is 0, we consider the test dataset to be a true Black Swan concerning the training and validation. The test dataset is not in the context of the training+validation, or vice versa (the training is not in the context of the test dataset). If the Visual Similarity Test is not 0, then the test dataset is in-context with the training.

H.11.3.4 Datasets Similarity Test and Venn Diagram Production

Below is a step-by-step description of how we compute the similarity values for two image datasets and then produce the corresponding Venn diagram:

1. **Computing Visual Signatures with Perceptual Hashing:**

   - The function **approach_4_visual_similarity(folder_path)** is used for each dataset.
   - It gathers all image files (with valid extensions such as .png, .jpg, etc.) from the given folder.
   - For each image, it opens the image using PIL and computes a perceptual hash (pHash) with the imagehash library. This pHash summarises the "visual essence" of the image.
   - The function returns a dictionary where the keys are the image filenames and the values are their corresponding pHash strings (unique signature for visual similarity).

2. **Indexing the Two Datasets:**

   - The code calls **approach_4_visual_similarity()** for Folder A and B separately to build two dictionaries: one for each dataset (folderA_dict and folderB_dict).
   - For example, folderA_dict might contain around 1054 images and folderB_dict around 8057 images, each entry mapping a filename to its pHash.

3. **Finding Duplicates:**

- The function **find_duplicates_in_dict()** is then used to compute visual duplicates within each dataset. It groups images by their hash (pHash in this case).
- This "duplicate analysis" produces maps (duplicatesA_map and duplicatesB_map) where any pHash that appears for more than one image results in a list of filenames.

4. **Comparing the Two Datasets:**

- The **compare_folders(folderA_dict, folderB_dict)** function compares the two datasets based on their hash signatures.
- Internally, it converts the hash values (pHash strings) for each dataset to sets (setA for Folder A, setB for Folder B). Then it computes the intersection between these two sets.
- The **intersection** represents the unique pHash values (i.e. visual features) found in both datasets. The number of elements in this intersection is stored as **unique_intersection_count**.
- In addition, the function builds lists of file pairs that have matching signatures (exact_matches) as well as filenames that appear only in one dataset (only_in_A and only_in_B). Below is an example of exact matches between experiments 1 and 4 training datasets:

5. **Calculating Unique Counts:**

- The number of unique visual signatures in Folder A is computed as **unique_A = len(set(folderA_dict.values()))**, and similarly for Folder B as **unique_B = len(set(folderB_dict.values()))**, in order to exclude any possible repetition.

6. **Generating the Venn Diagram:**

- The **generate_venn_diagram()** function is called.
- Inside the function, the code calculates the numbers for A-only and B-only by subtracting the **unique_intersection_count** from **unique_A** and **unique_B**, respectively.
- The venn2 function from the **matplotlib_venn** library is then used to plot a Venn diagram with:
    - Left circle representing Dataset A's unique signatures,
    - Right circle representing Dataset B's unique signatures,
    - The overlapping area representing **unique_intersection_count**.

- A title is constructed that shows key metrics (total images, unique counts, intersection, the IoU computed as 100*(unique_intersection_count/(unique_a+unique_b-unique_intersection_count)) formatted to two decimal places, and the no-dup coverage percentages).
- Finally, the plot is saved to a file path that is built using the **output_report_directory** parameter so that the Venn diagram is saved in the specified location.

H.11.3.5 Scope Limitations

The current experiments do not address:

- Sensitivity to threshold variations for detection confidence and bounding box overlap.
- Real-world, dynamically acquired operational data.
- Effects of image pre-processing or data augmentation techniques.

These are planned future extensions of this work to further validate and refine the insights gained from this study.
Overall, this experiment design serves as a foundation for evaluating how context-aware data curation strategies influence ML model robustness, particularly in safety-critical domains where both typical and unforeseen (Black Swan) scenarios are of concern.

H.11.3.6 Experiment results

The following are the results of the experiments above. We arranged according to the performance over Black Swan test dataset:

Table H.58 Experiment results

Experiment	**Training Strategy**	**Validation Strategy**	Valid **mAP50**	**Testing Strategy**	Test **mAP50**
Group 1: Experiments with OOC training.					
Experiment 1: see link **Total:** 9495 images	**Total:** 6753(71%) Random, OOC: 6753 (100%)	**Total:** 1679 (18%) Random, OOC: 1679 (100%)	97	**Total:** 1063 (11%) Random, OOC: 1063 (100%)	97

Experiment 4: see link **Total:** 10658 images	**Total:** 8000 (75%) In-context images: 0 OOC images: 8000 (100%)	**Total:** 1500(14%) In-context images: 0 OOC images: 1500 (100%)	90	**Total:** 1158 (11%) In-context, non-typical, Black Swan (batch scenarios a) operational images: 1158 (100%)	24
Experiment 5: See link **Total:** 14384 images	**Total:** 9924(69%) In-context images: 0 OOC images: 9924 (100%)	**Total:** 1399(10%) In-context images: 0 OOC images: 1399 (100%)	95	**Total:** 3061 (21%) In-context typical operations images: 3061 (100%)	93
Group 2: Experiments with in-context, typical operations coverage.					
Experiment 6: See link **Total:** 14479 images	**Total:** 10015 (69%) **In-context typical operations images:** 3270 (32%) **OOC situations:** 6745 (67%)	**Total:** 1403(10%) **In-context typical operations images:** 1202 (85%) **OOC situations:** 201 (15%)	99.5	**Total:** 3061(21%) **In-context typical operations images**: 3061 (100%)	99
Experiment 7: See link **Total:** 12578 images	**Total:** 10015 (80%) **In-context typical operations images**: 3270 (32%) **OOC situations:** 6745 (67%)	**Total:** 1405(11%) **In-context typical operations images**: 1202 (85%) **OOC situations:** 201 (15%)	99.5	**Total:** 1158(9%) **In-context, non-typical, Black Swan operational images:** 1158 (100%)	13

Experiment 10: See link **Total:** 101548 images	**Total:** 85677 (84%), In-context typical operations images: 3270 (4%) OOC situations: 82257 (96%)	**Total:** 14713 (14%), In-context typical operations images: 1202 (8%) OOC situations: 13511 (15%)	93	**Total:** 1158 (1%) In-context, non-typical, Black Swan (batch scenarios a) operational images: 1158 (100%)	35
Group 3: Experiments with black swan coverage and random images.					
Experiment 2: see link **Total:** 9495 images	**Total:** 6753 (71%) In-context, Black Swans (batch scenarios a): 224 (3.3%) Random, OOC: 6529 (96.7%)	**Total:** 1679 (18%) Random, OOC: 1679 (100%)	97	**Total:** 1063(11%) Random, OOC: 1063 (100%)	97
Experiment 3: See link **Total:** 10658 images **Experiment 8.1:** See link **Total:** 12561 images	**Total:** 8000 (75%) In-context non-typical black swan (batch scenarios A) images: 2390 (30%) OOC images: 5610 (70%) **Total:** 8000 (64%) **In-context non-typical black swan** (batch scenarios A) **images:** 2390 (30%) **OOC images:** 5610 (70%)	**Total:** 1500 (14%) In-context non-typical black swan (batch scenarios A) images: 701 (47%) OOC situations: 799 (63%) **Total:** 1500(12%) **In-context non-typical black swan** (batch scenarios A) **images:** 701 (47%) **OOC situations:** 799 (63%)	82 84	**Total:** 1158 (11%) In-context, non-typical, Black Swan (batch scenarios a) operational images: 1158 (100%) **Total:** 3061(24%) **In-context typical operations images**: 3061 (100%)	67 97

Experiment 8.2: See link **Total:** 10658 images	**Total:** 8000 (75%) **In-context non-typical black swan** (batch scenarios A) **images:** 2390 (30%) **OOC images:** 5610 (70%)	**Total:** 1500 (14%) **In-context non-typical black swan**(batch scenarios A) **images:** 701 (47%) **OOC situations:** 799 (63%)	83	**Total:** 1158 (11%) **In-context, non-typical, Black Swan** *(batch scenarios B)* **operational images:** 1158 (100%)	28
Group 4: Experiments with all types in, black swans, typical + out/context coverage and random images.					
Experiment 9.1: See link **Total:** 22374 images	**Total Training:** 14961 (67%) **In-context non-typical black (batch A) swan images.** and **In-context typical operations images**: 5655 (37%) **OOC images:** 9306 (63%)	**Total Valid:** 2903 (13%) **In-context non-typical black swan (batch A) images.** and **In-context typical operations images**: 1911(67%) **OOC images:** 992 (34%)	91	**Total Test:** 4510 (20%) **In-context typical operations images**: 3461 (77%) **OOC images:** 1049 (23%)	99
Experiment 9.2: See link **Total:** 23532 images	**Total Training:** 17196 (73%) **In-context non-typical black swan images.** and **In-context typical operations**	**Total Valid:** 5178 (22%) **In-context non-typical black swan images.** and **In-context typical operations**		**Total Test:** 1158 (5%) **In-context non-typical black swan images:** 1158 (100%)	40%

	images: 7851 (45%) **OOC images:** 9502 (55%)	**images**: 3165 (61%) **OOC images:** 2079 (39%)			
Experiment 9.3: See link **Total:** 57924 images	**Total Training:** 51588 (89%) **In-context non-typical black swan (batch A) images.** and **In-context typical operations images**: 7851 (45%) **OOC images:** 9502 (55%)	**Total Valid:** 5178 (9%) **In-context non-typical black swan (batch A) images.** and **In-context typical operations images**: 3165 (61%) **OOC images:** 2079 (39%)		**Total Test:** 1158 (2%) **In-context non-typical black swan (batch B) images:** 1158 (100%)	
Experiment 9.4: See link **Total:** 23533 images	**Total Training:** 16788 (71%) **Black Swan Batch B:** 695(4.1%). **Black Swan Batch A:** 1866 (11.11%). **In-context typical operations images**: 3273 (19.49%). **OOC images:** 10954 (65.24%)	**Total Valid:** 4426 (19%) **Black Swan Batch B:**231(5.22%). **Black Swan Batch A:**622(14.05%). **In-context typical operations images**: 1604(36.24). **OOC images:**1969 (44.49%)	98	**Total Test:** 2319 (10%) **Black Swan Batch B:**232(10%). **Black Swan Batch A:** 622(26.82%). **In-context typical operations images**: 1197(51.61%). **OOC images:** 268 (11.55%)	44% 99

Throughout these experiments, we hypothesised various outcomes concerning black-swan coverage, dataset size, and overall composition (in-context vs. random). Below, we review how actual results aligned with or diverged from those expectations, highlight any unanticipated correlations, and discuss the implications for testing strategy, particularly in safety-critical domains. To validate the comparability of the datasets, we used Python code to compare each dataset with another, both of which were downloaded directly from their respective Roboflow repositories. This comparison indicates their similarity, meaning they contain identical images across experiments.

Note 1: the Venn diagrams only show unique hashes of images. Therefore, two similar-sized datasets can be different in terms of the number of unique images. The hashes give us a verifiable indication of a dataset's true diversity. The more unique hashes in a dataset, the more diverse it is (in our soft opinion).

Note 2: We used the imagehash library to estimate the likelihood of similarity. The library focuses on visual similarity, not bit-for-bit. We don't use this method to catch copies of the same images in the same dataset, but rather to determine how visually similar an image is to another, yet different. Also, to test visual similarities accurately among datasets, we excluded bit-by-bit similar images. The number of bit-by-bit similar images in some datasets is negligible and does not change the results of the experiments. However, they may have an impact on the visual similarity IoU.

H.11.4 Group 1: OOC only.

In this group of experiments, we will examine how excluding deliberate black swan scenarios from the training or validation will impact model performance in random scenario testing, typical operations, and how such a strategy will affect our trust in relying solely on random images or typical operations performance for claims of robustness during Black Swan scenarios.

(i) Result of Experiment 1

exp1_testing dataset: 854 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of approximately 33.87%.

exp1_train+valid dataset: 5051 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of roughly 61.09%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 576 unique signatures are common to both datasets.

A shared visual distribution likelihood (unique pHash datasets IoU) between exp1_testing and exp1_train+valid is 10.8%. The IoU is >5% of the similarity threshold. Therefore, the datasets are potentially in distribution with each other.

Comparing dataset without in-dataset duplications (considering only one hash number if repeated multiple times):

- exp1_testing -> exp1_train+valid: (576 / 854) ≈ 67.45%
    - It means roughly 67% of Exp1-testing is random InD images with training+validation.
    - While 33% of exp1_testing is random OOD images.

exp1_train+valid -> exp1_testing: (576 / 5051) ≈ 11.40%.

ML robustness test consolidation claim 1: The performance indicates that, given OOC training and validation, the model can generalise 97% of the time over OOC scenarios that are InD and OOD with training and validation.

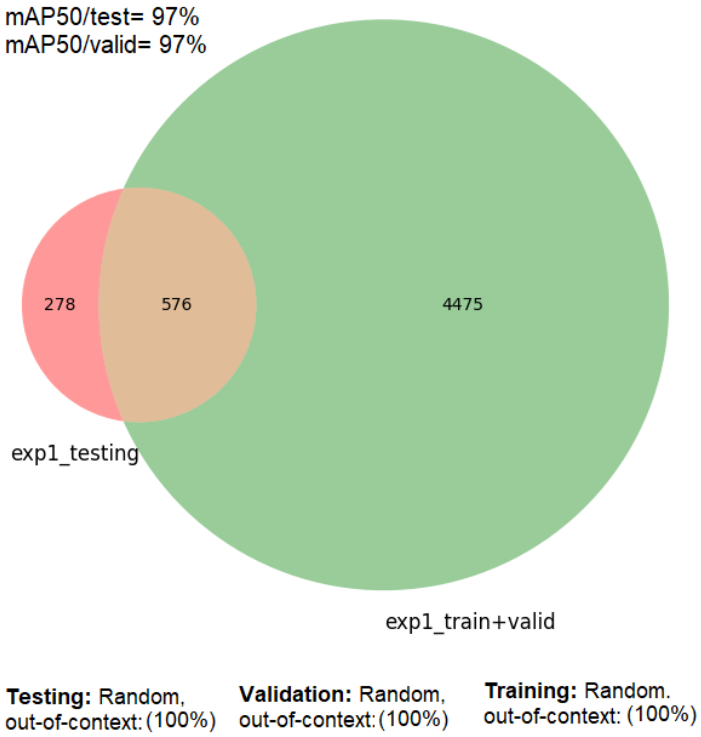


Figure H.45 Exp.1 ML training and testing strategy performance

(ii) Results of Experiment 4

exp4_testing dataset: 1158 images, 763 unique, with a visual similarity (within the dataset) rate of approximately 44.13%.

exp4_training+validation dataset: 9457 images, 5554 unique, with a visual similarity (within the dataset) rate of roughly 63.50%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 0 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between exp4_testing and exp4_training+validation: 0.0%

No-dup coverage (comparing dataset without in-dataset duplications (considering only one hash number if repeated multiple times):

exp4_testing -> exp4_training+validation: (0 / 763) ≈ 0.00%.

exp4_training+validation -> exp4_testing: (0 / 5554) ≈ 0.00%.

- The test dataset is true out of distribution with training and validation combined.

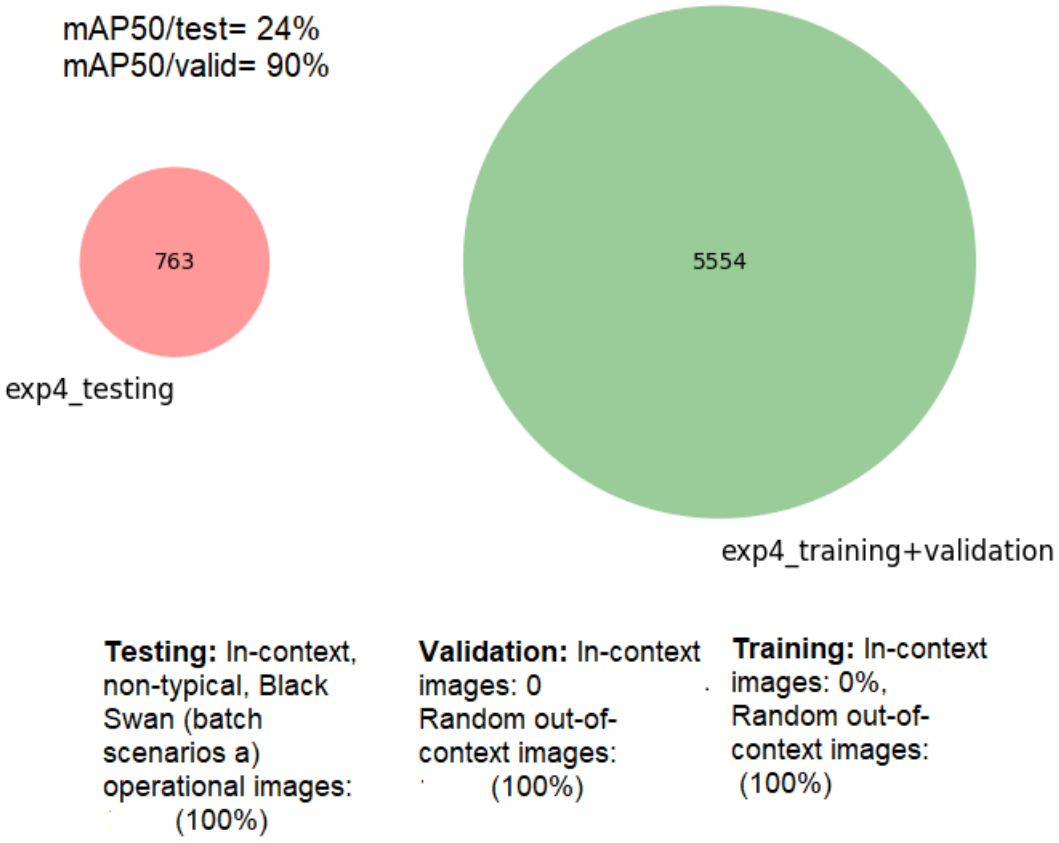


Figure H.46 Exp.4 ML training and testing strategy performance

ML robustness test consolidation claim 4: The performance indicates that, given OOC training and validation, the model can generalise 24% of the time over in-context, OOD Black Swan scenarios that are OOD with training and validation.

(iii) Comparing Exp.1 and 4

Exp.1 and 4 training and validation:

exp1_train+valid dataset: 5051 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of approximately 65.60%.

exp4_train+valid dataset: 5554 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of roughly 63.50%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 4891 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between exp1_train+valid and exp4_train+valid: 85.59%

comparing dataset without in-dataset similarities (considering only one hash number if repeated multiple times):

exp1_train+valid -> exp4_train+valid: (4891 / 5051) ≈ 96.83%.

exp4_train+valid -> exp1_train+valid: (4891 / 5554) ≈ 88.06%.

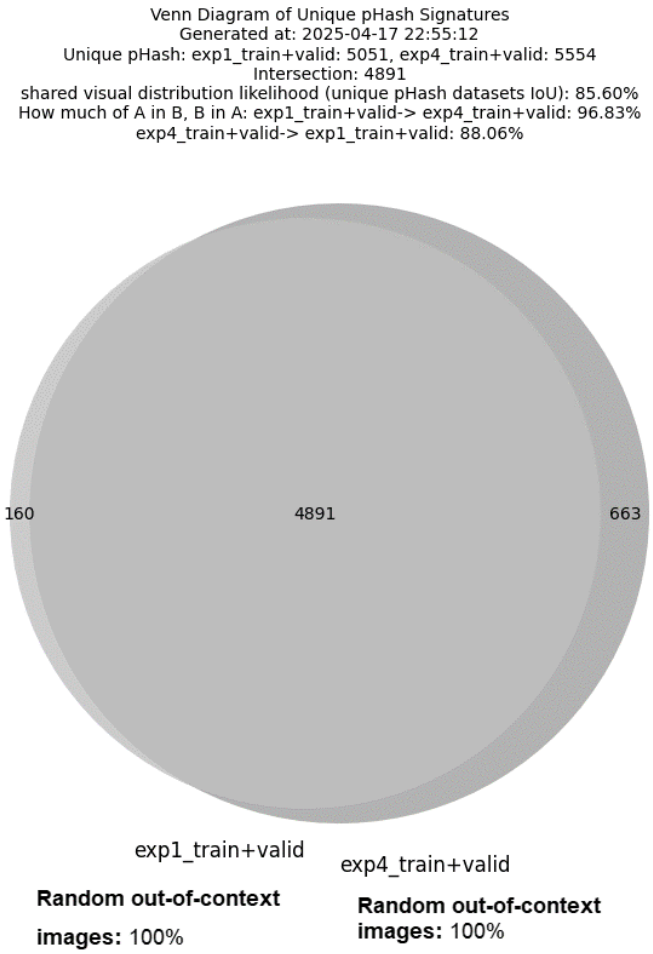


Figure H.47 Comparing the training strategy similarity between Exp.1 and 4.

Notice: Exp.1 and 4 validation results showed an expected similarity of performance of (indicating map50 90%+). While testing on in-context black swans did not improve. Thus affirming our suspicion that good performance over random images does not necessitate good performance in in-context black swan scenarios. Therefore, we do need to derive Black Swan scenarios and make a separate test dataset for testing over them.

Experiment 1 and 4 testing:

exp1_testing dataset: 1054 images, 854 unique, with a visual similarity (within the dataset) rate of approximately 33.87%.

exp4_testing dataset: 1158 images, 763 unique, with a visual similarity (within the dataset) rate of roughly 44.13%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 0 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between exp1_testing and exp4_testing: 0.0%

No-dup coverage (comparing dataset without in-dataset duplications (considering only one hash number if repeated multiple times):

exp1_testing -> exp4_testing: (0 / 854) ≈ 0.00%.

exp4_testing -> exp1_testing: (0 / 763) ≈ 0.00%.

- The similarity test confirms that the Exp.1 and Exp.4 test datasets are entirely different. The Exp. 1 test is an OOC random imaged dataset, while Exp.4 is an in-context Black Swan. This is a useful indication of the robustness of the similarity test using imagehash.

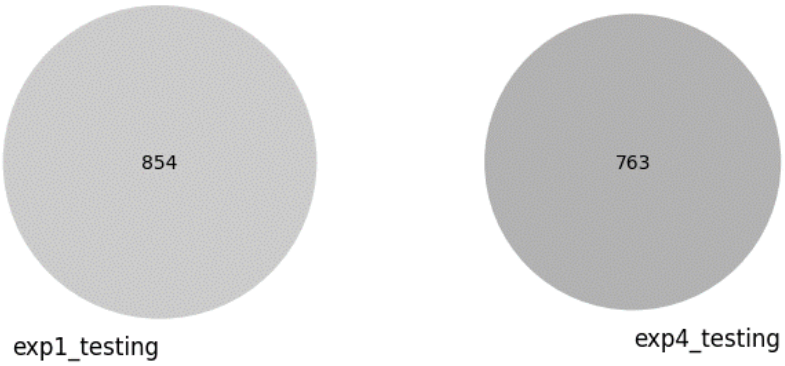


Figure H.48 Comparing the similarity of testing strategies between Exp.1 and 4. Exp.1 is random, 4 includes Black Swan only.

Table H.59 Expected behaviour and results for exp.1 and 4

Behaviour over the testing dataset hypothesis	**Outcomes of the test**
Exp.1 trained, validated on random, tested on random. Exp.4 trained, validated on random, tested on Black Swan. $Exp1_{mAP50/test} \approx Exp4_{mAP50/test}$	$Exp1_{mAP50/test} = 97\%$ $Exp4_{mAP50/test} = 24\%$ Therefore: $Exp1_{mAP50/test} > Exp4_{mAP50/test}$

	Conclusion 1: Training with OOC images alone may not assure performance over Black Swan scenarios.

Finding 1:

Invalidation of Claim 1: ML robustness test consolidation Claim 1 asserted that the model should generalise well over OOD scenarios according to the model's performance over training and validation scenarios. ML robustness test consolidation claim 4 had invalidated the claim in 1. Thus, demonstrating that the dataset approach is inadequate.

Stage 4 has helped us to predict a Black Swan scenario, which is also OOD, where the model's performance had been invalidated. This means:

- The training and testing approach over OOC images does not assure consistent performance over Black Swan scenarios and data shift.
- Stage 4 has helped us predict an unmitigated data shift scenario where the diversity of choosing the datasets 1 or 4 cannot handle. This means the datasets are not trustworthy to enable a reliable model.

(iv) Exp.5 results

In this experiment, we used a dataset of random images and tested them on in-context typical operations. The training and validation dataset is partially the same as Exp. 4.

exp5_testing dataset: 3061 images, 2964 unique, with a visual similarity (within the dataset) rate of approximately 5.06%.

exp5_training+validation dataset: 11211 images, 7924 unique, with a visual similarity (within the dataset) rate of roughly 44.02%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 0 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between exp5_testing and exp5_training+validation: 0.0%

No-dup coverage (comparing dataset without in-dataset duplications (considering only one hash number if repeated multiple times):

exp5_testing -> exp5_training+validation: (0 / 2964) ≈ 0.00%.

exp5_training+validation -> exp5_testing: (0 / 7924) ≈ 0.00%.

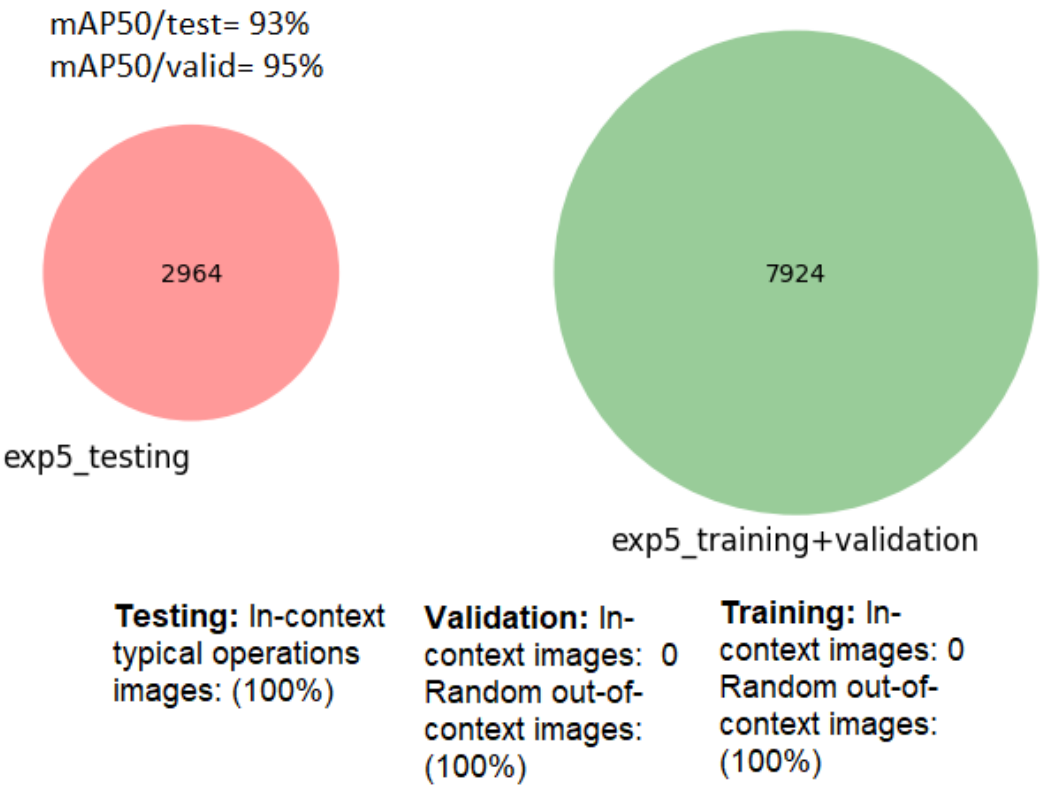


Figure H.49 Similarity test between training and training+validation for Exp.5

ML robustness test consolidation claim 5: The performance indicates that, given OOC training and validation, the model can generalise 93% of the time over in-context, OOD, and Typical scenarios. This means that the diversity of unique examples is sufficient to manage in-context, typical operation data shift.

Note: The test dataset is entirely out of distribution from the training and validation combined. This is not a Black Swan dataset. This raises the question of how, in terms of dataset similarity, we make sense of images or datasets that are relatively novel but typical operations. Although the testing dataset is out of distribution, we expect lower performance from the model in those scenarios.

We saw this in experiment 4, where the testing and training datasets were 0% similar. The model performed poorly, validating that the testing datasets are truly Black Swans. However, in this case (exp.5), the testing dataset is 0% similar to the training; however, the model generalised well.

(v) Comparing Exp.4 and 5

We will compare Exp.5 and 4 datasets to consolidate the latest realisation. Let's have a look at how the training and validation of exp.4 and 5 similarities tell us about how closely related their training is:

Exp4_training dataset: 7957 images, 4963 unique, with a visual similarity (within the dataset) rate of approximately 59.56%.

Exp5_training dataset: 9813 images, 6738 unique, with a visual similarity (within the dataset) rate of approximately 47.52%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 3882 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between Exp4_training and Exp5_training: 49.64%

comparing dataset without in-dataset similarities (considering only one hash number if repeated multiple times):

Exp4_training -> Exp5_training: (3882 / 4963) ≈ 78.22%. This tells us that nearly 79% of situations in exp.4 are included in exp.5, so perhaps this is why there is an improvement over OOD. We need to investigate this, which we do in exp.7.

Exp5_training -> Exp4_training: (3882 / 6738) ≈ 57.61%.

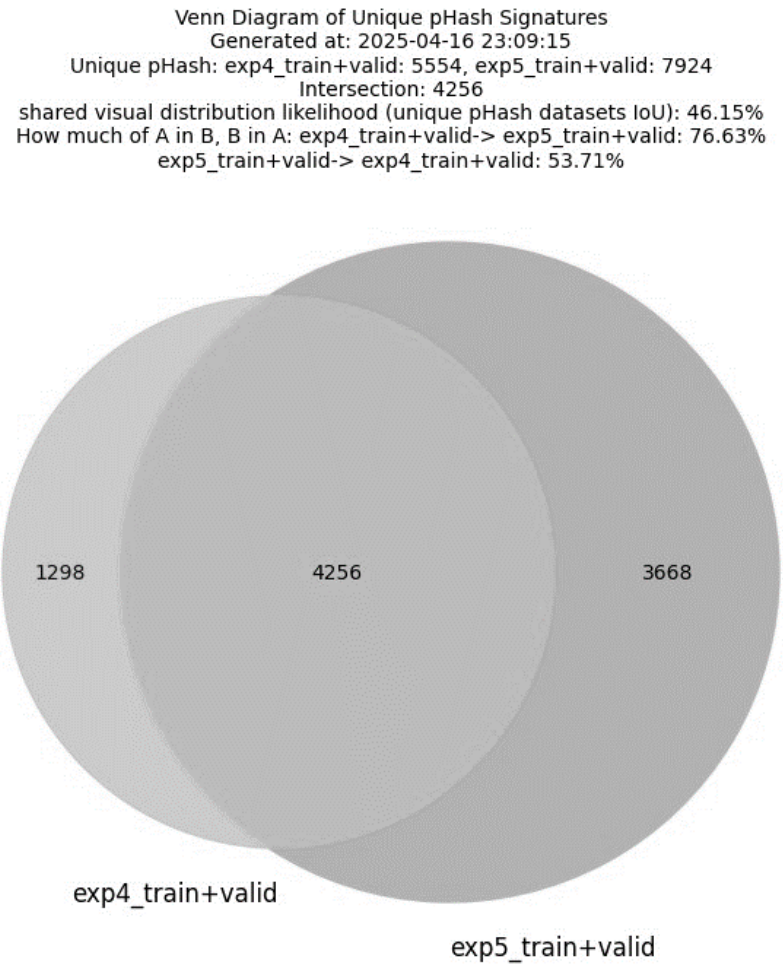


Figure H.50 Comparing the similarities between Exp.4 and 5 training and validation

Experiment 5 training and validation appear to be partially in distribution with exp.4 training and validation (IoU > 5%), with roughly 77% of exp.4 training validations present in exp. 5 training and validation. The size of the Exp.5 training strategy includes approximately 2370 more unique

instances. In this case, the improvement over typical operations' novelty could be attributed to increased training instances. Note that the exp5_testing is OOD with We need to confirm this by enhancing the experiment 5 training and validation, which we will address in experiment 10.

H.11.5 Group 2: Training with in-context typical operations coverage.

In this group of experiments, we will examine how typical operations scenarios will impact performance over Black Swan scenarios or similar scenarios.

(i) Exp.10 results

exp10_testing dataset: 1158 images, 762 unique, with a visual similarity (within the dataset) rate of approximately 44.04%.

exp10_training+validation dataset: 100390 images, 45917 unique, with a visual similarity (within the dataset between training and validation) rate of roughly 66.97%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 0 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between exp10_testing and exp10_training+validation: 0.0%

No-dup coverage (comparing dataset without in-dataset duplications (considering only one hash number if repeated multiple times):

exp10_testing -> exp10_training+validation: (0 / 762) ≈ 0.00%.

exp10_training+validation -> exp10_testing: (0 / 45917) ≈ 0.00%.

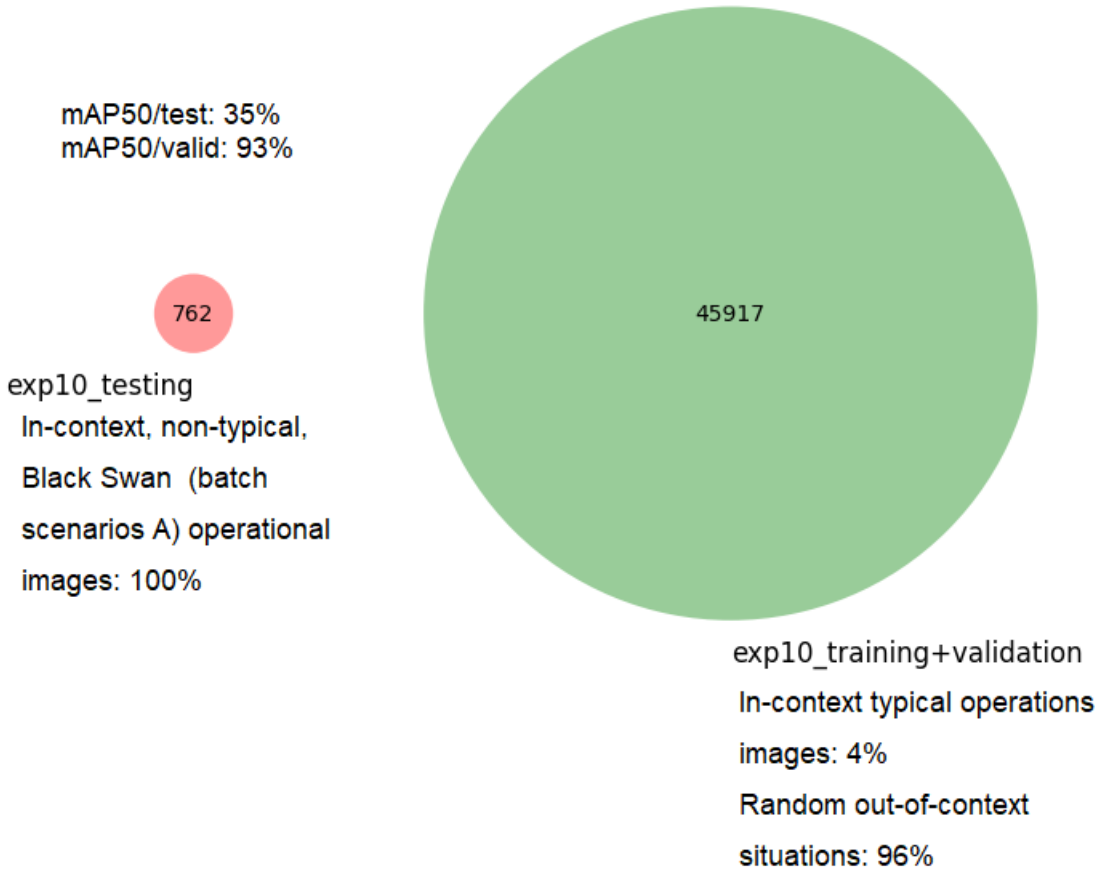


Figure H.51 Exp.10 similarity test

ML robustness test consolidation claim 10: The performance indicates that, given in-context, typical operations and OOC training and validation, the model can generalise 35% of the time over in-context, OOD, non-typical scenarios that are OOD with training and validation. **This means:** the diversity of unique examples is insufficient to manage in-context, non-typical operation data shift.

(ii) Comparing Exp.10 and 4

exp4_train+valid dataset: 5554 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of approximately 63.50%.

exp10_train+valid dataset: 45917 unique perceptual hashes found in this dataset, with a visual similarity (within the dataset) rate of roughly 66.97%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 5148 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between exp4_train+valid and exp10_train+valid: 11.11%

comparing dataset without in-dataset similarities (considering only one hash number if repeated multiple times):

exp4_train+valid -> exp10_train+valid: (5148 / 5554) ≈ 92.69%.

exp10_train+valid -> exp4_train+valid: (5148 / 45917) ≈ 11.21%.

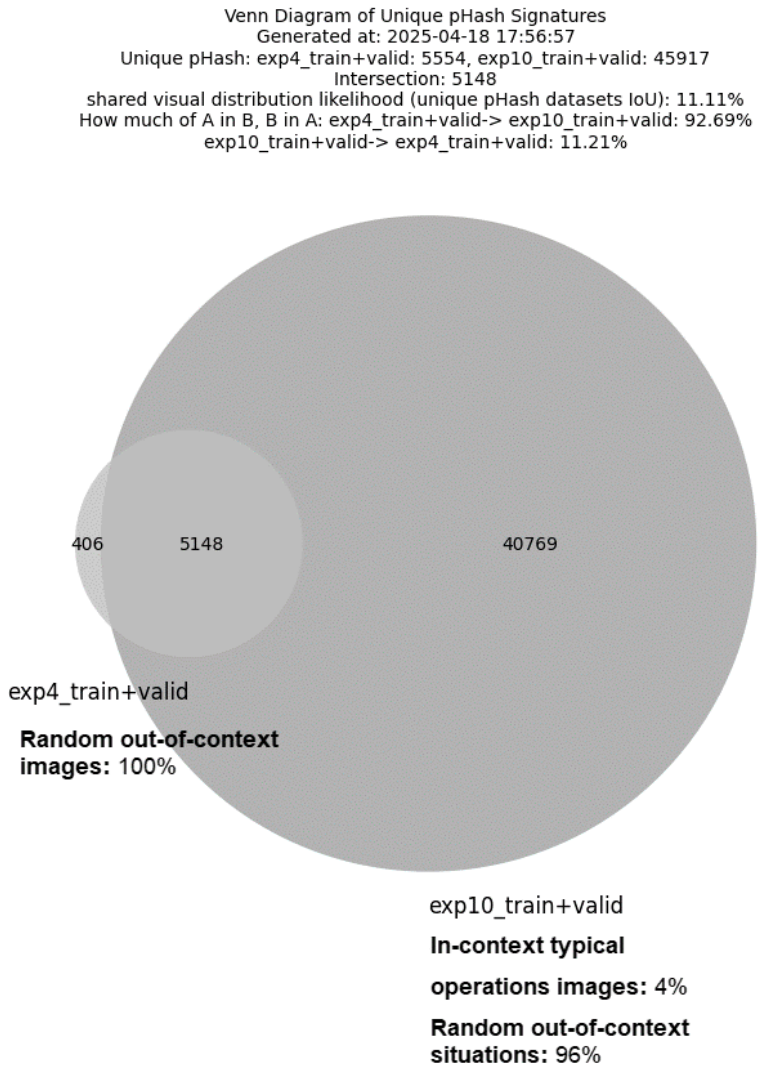


Figure H.52 Exp.10 and 4 similarity

By increasing the number of OOC images in the training and validation of exp.10, we were expecting a notable increase in performance over Black Swan scenarios. However, exp. 4 showed that random images alone are not enough to cover for Black Swans. The results showed insignificant improvement, given that we added an in-context typical operations dataset into exp.10.

Table H.60 Comparing exp.10 and 4

Behaviour over the testing dataset hypothesis	**Outcomes of the test**
$Exp10_{mAP50/test} \gg Exp4_{mAP50/test}$ Both experiments were tested on the same Black Swans scenarios (batch A)	$Exp4_{mAP50/test} = 24\%$ $Exp10_{mAP50/test} = 35\%$

	Therefore: $Exp10_{mAP50/test} > Exp4_{mAP50/test}$ **Conclusion 6:** A significant increase in dataset diversity resulted in a marginal increase in performance. This means we need more diversity, which means the Black Swans batch A are a true Black Swan scenario (data shift) relative to the training and validation dataset. This means an increase in the number of OOC images does not necessarily mitigate black swans. We need Black Swan scenarios to be present in the dataset's training and validation.

(iii) Comparing Exp.10 and 5

We used the dataset in exp.5, which is considerably similar to the exp.10 dataset (roughly 82% of exp.5 training and validation hashes in exp.10). The validation of exp.10 is composed of in-context typical operations and random images. The model performed well (93%) on such a composition, which indicates that if we tested the model on a similar test set as exp.5 (only in-context typical operations), we can be confident it would have produced comparable results.

Now, since we used most of the exp5 training dataset in the exp.10 dataset training, it is plausible to expect improved performance in in-context scenarios, especially since exp.5 showed that the dataset enabled the model to perform well on an the OOD test set.

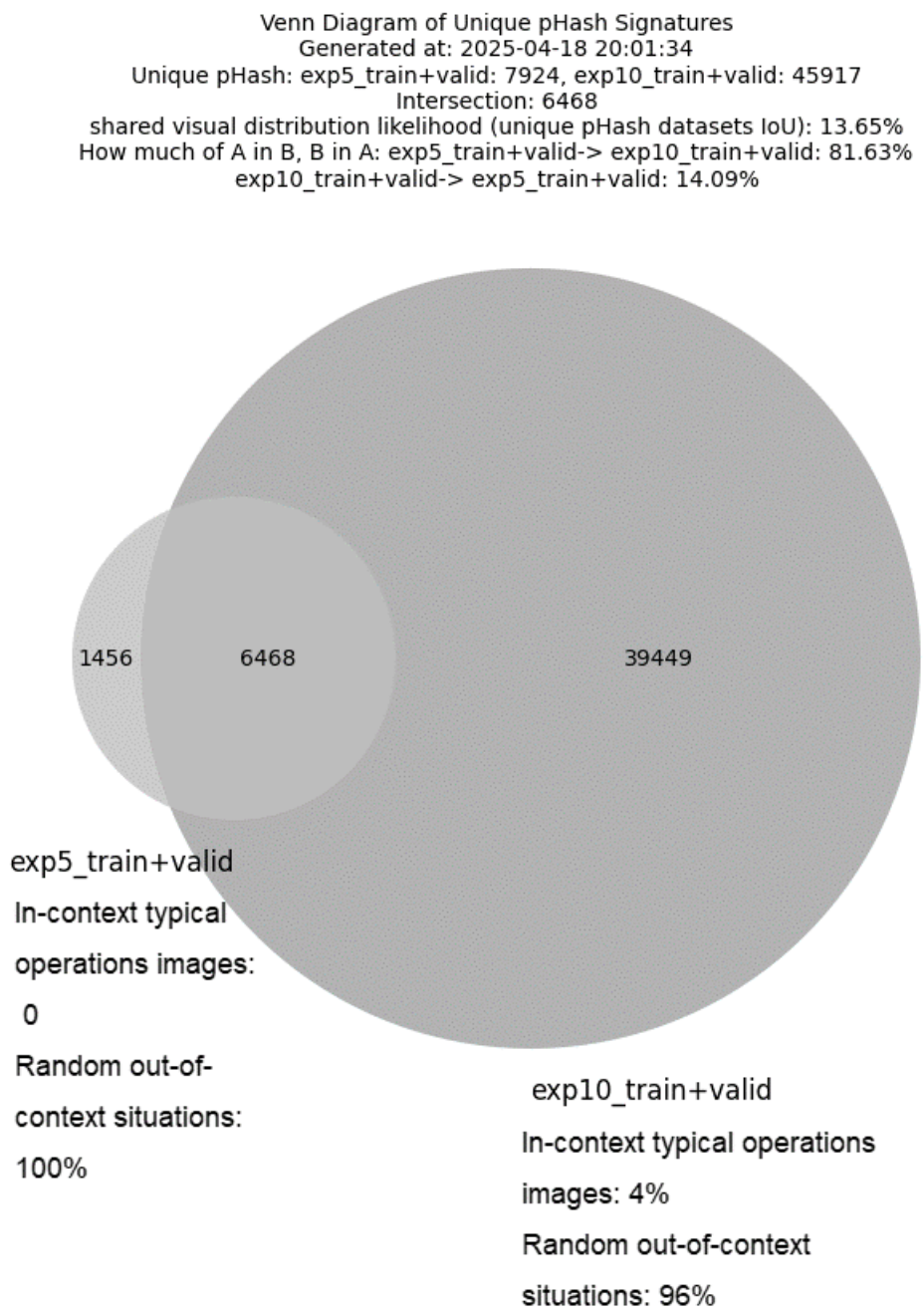


Figure H.53 Comparing exp.10 training with exp.5 training datasets

In Exp.10 we increased the size of training and validation, compared to Exp.5, by nearly 38000 unique situations (each hash is a situation which may have multiple visually the same images) of training and validation examples. We also injected the Exp.10 training and validation with nearly 4472 images of in-context, typical operations data (see the composition of Exp.10 training strategy in the figure above). In theory, we expected that if we train the same model architecture (in this case, Roboflow 3), with such an improved dataset, intuitively, we should have a better, if not the same, performance over another OOD. The question we naturally asked ourselves:

> Since training on random images with a certain diversity of examples (unique pHashes) has produced a model that generalises well on OOD in-context typical operations, increasing the number of unique examples (unique pHashes) and including in-context

typical operations should result in better performance across different sets of OOD testing datasets.

For our hypothesis to be consistent, we must demonstrate that testing datasets for Exp.5 and Exp.10 are entirely different, even though they are in context. Below is a demonstration of how dissimilar they are:

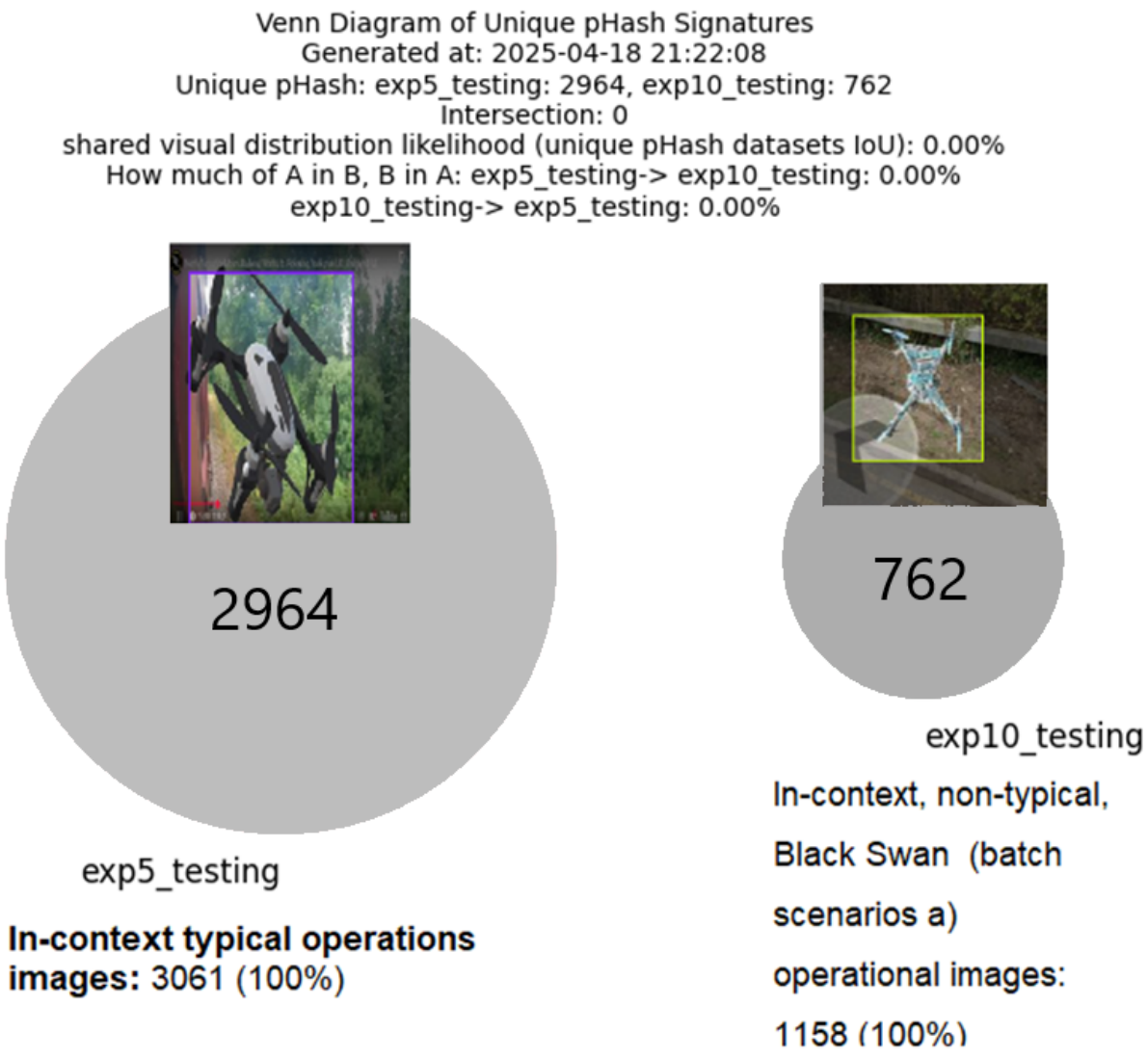


Figure H.54 Examples testing images, right hand showing Black Swan situation, left hand showing typical operation.

We are in a situation where we trained the same model, one on random images with a smaller dataset, while the other was mixed with in-context and random images, using a larger dataset. Exp.10 testing yielded a mAP50 of 35%, indicating inferior performance.

This confirms that the OOD testing dataset for Exp.10 is a true Black Swan and that Stage 4 process has helped us discover an unmitigated Black Swan scenario. It also means that testing only on random images and typical operations is insufficient; we need Black Swan scenarios and as many as possible to ensure that the training strategies provide us with a reliable model. Stage 4 had helped us to discover **Training Strategy-Unmitigated Operational Data Shift.**

Finding 2: We realised that the "discrepancy" (where we have two OOD test sets, one was easier than the other for the model) we see isn't a bug, it's precisely the signal we want: So, in the context of Black-Swan scenario discovery, mitigation and validation process:

- Given an accepted threshold of detection confidence in-context of a CuneiForm, we've hit **Training Strategy-Unmitigated Operational Data Shift**, **an unmitigated relative Black Swan scenario or situation** when the model knocks down performance below the accepted threshold over an image or the set of images' contexts that are outside the distribution of the training and validation. "relative" means relative to the model's training and validation.
    - CuneiForm + 0% pHash overlap + lower mAP ⇒ Confusing Data Shift (Unmitigated Black Swan)
    - The exp4_testing dataset (Black Swan CuneiForms: H.36, 37, 38, 39, 40, 41) is, in fact, relative unmitigated data shifts that had been predicted by stage 4.
    - In other words, if we choose exp4_training+validation dataset, the diversity and the size of such a dataset des not mitigate the Black Swan Cuneiforms.
- Given an accepted threshold of detection confidence in-context of a CuneiForm, when ML performance stays higher than the threshold over the tested image or dataset outside the distribution of the training and validation, we've hit a **Training Strategy-Mitigated Operational Data Shift, a mitigated relative Black Swan situation or scenario,** surprise or shift relative to what had been learned about the real world.
    - CuneiForm + 0% pHash overlap + higher mAP ⇒ Normal Operational Data Shift (mitigated Black Swan(.

H.11.6 Group 3: Training with in-context Black Swan (data shifts) coverage.

In this batch of experiments, we will include Black Swans in the training and validation and check how this improves robustness.

(i) Exp. 3 dataset results

exp3_test dataset: 1157 images, 764 unique, with a visual similarity (within the dataset) rate of approximately 44.08%.

exp3_train+valid dataset: 9489 images, 6969 unique, with a visual similarity (within the dataset) rate of roughly 41.57%.

Note: Visual similarity does not mean exact copies. Common unique signatures: 57 unique signatures are common to both datasets.

shared visual distribution likelihood (unique pHash datasets IoU) between exp3_test and exp3_train+valid: 0.74%

No-dup coverage (comparing dataset without in-dataset duplications (considering only one hash number if repeated multiple times):

exp3_test -> exp3_train+valid: (57 / 764) ≈ 7.46%.

exp3_train+valid -> exp3_test: (57 / 6969) ≈ 0.82%.

- The test dataset is potentially out of distribution with training and validation combined.

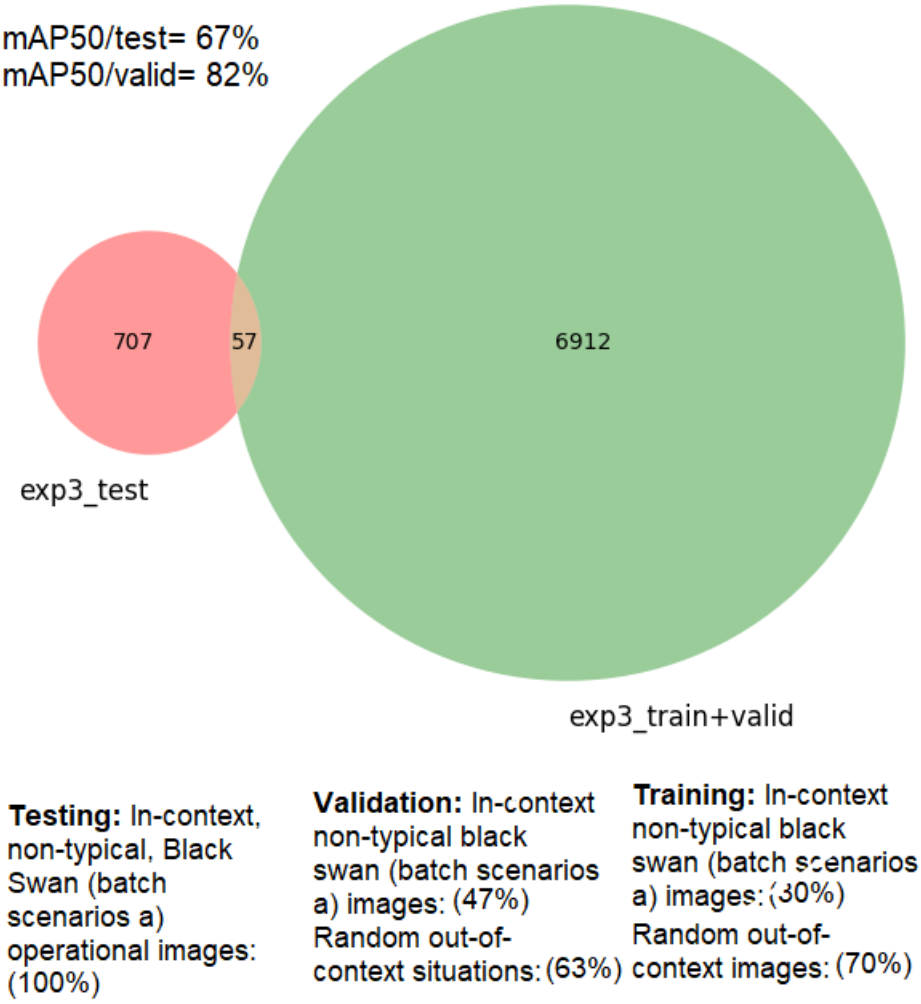


Figure H.55 Exp.3 development

H.11.1 Group 4: All in

H.11.2 Conclusions

Overall, the experiments largely supported the initial expectations that (1) black swan coverage is pivotal to black swan performance and (2) scaling data size plus contextual alignment improves reliability. The unexpected or underestimated outcome was just how drastically performance can drop on purely black swan tests (e.g., from near 100% on standard data to under 20% on black swans) unless explicit coverage is provided in training. This outcome highlights both the necessity

of a stand-alone black swan test partition and the importance of methodically increasing black swan coverage for a safer, more trustworthy model.

The results of both development processes confirm the positive impact of incorporating Black Swan-enhanced datasets on the robustness of object detection models when facing unforeseen scenarios. This shows that if we train a model from a random collection of images taken from a public domain or any form of dataset gathering, we cannot trust it to perform well on Black Swan scenarios related to the operational domain of concern, like the Train Track zone. We will need to predominately include in-context training scenarios (especially within Black Swan scenarios) and validate them too much for us to have better trust in the model. Extra images can used for improvement on robustness but not to justify claims in the safety case about suitability for the specific operational domain. The following is the summary of the findings:

1. Validation within an ML development environment confirmed the effectiveness of Black Swan-enhanced training across different testing conditions.
2. These findings strengthen the argument for using Black Swan scenario validation as an assurance process to evaluate a model's adaptability and reliability in real-world safety-critical applications, which appears to be useful evidence to back up safety case claims of model suitability.

H.11.2.1 Implications for Future Research and Applications

The experiments were made up to believe that Black Swan-enhanced datasets should be systematically integrated into ML training workflows for safety-critical perception systems. Additional experiments could explore the impact of synthetic Black Swan data augmentation (e.g., AI-generated variations) on model generalisation. Finally, regulatory bodies and Safety assurance frameworks (e.g., SACE, AMLAS) should consider Black Swan validation as a key component in assessing Ai reliability for real-world deployment.

H.11.3 Summary of all Experiments

Table H.61 Summary of all experiments

Experiment	**Training Strategy**		**Testing strategy**
	Training Dataset	**Validation Dataset**	**Testing Dataset**
	Total: 6753(71%)	**Total:** 1679 (18%)	**Total:** 1063 (11%)

<table>
<tr><td rowspan="3">Experiment 1:

see link

Total: 9495 images</td><td>Random, OOC: 6753 (100%)</td><td>Random, OOC: 1679 (100%)</td><td>Random, OOC: 1063 (100%)</td></tr>
<tr><td>No CuneiForms</td><td>No CuneiForms</td><td>No CuneiForms</td></tr>
<tr><td colspan="3">Note: The validation and test sets used in this experiment are identical to those in Experiment 2. The random images in the training set are identical to those in the training set used in Experiment 2. No pre-processing steps were applied. No augmentations were applied.</td></tr>
<tr><td rowspan="3">Experiment 2:

see link

Total: 9495 images</td><td>Total: 6753 (71%)</td><td>Total: 1679 (18%)</td><td>Total: 1063(11%)</td></tr>
<tr><td>In-context, Black Swans: 224

Random, OOC: 6529</td><td>Random, OOC: 1679 (100%)</td><td>Random, OOC: 1063 (100%)</td></tr>
<tr><td>Black Swans A CuneiForm: H.36</td><td>No CuneiForms</td><td>No CuneiForms</td></tr>
<tr><td colspan="3">Note: The validation and test sets are the same as those in Experiment 1. The random images in the training set are identical to those in Experiment 1's training set. No pre-processing steps were applied. No augmentations were applied.</td></tr>
<tr><td rowspan="4">Experiment 3:

See link

Total: 10658 images</td><td>Total: 8000 (75%)</td><td>Total: 1500 (14%)</td><td>Total: 1158 (11%)</td></tr>
<tr><td>In-context non-typical black swan images: 2390 (30%)

OOC images:

5610 (70%)</td><td>In-context non-typical black swan images: 701 (47%)

OOC situations:

799 (63%)</td><td>In-context, non-typical, Black Swan operational images:

1158 (100%)</td></tr>
<tr><td colspan="3">Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td></tr>
<tr><td colspan="3">Note: The test dataset is identical to the one used in Experiment 4. The instantiations of the cuneiforms are different in each dataset. No pre-processing steps were applied. No augmentations were applied.</td></tr>
<tr><td rowspan="2">Experiment 4:

see link

Total: 10658 images</td><td>Total: 8000 (75%)</td><td>Total: 1500(14%)</td><td>Total: 1158 (11%)</td></tr>
<tr><td>In-context images: 0

OOC images:

8000 (100%)</td><td>In-context images: 0

OOC images:

1500 (100%)</td><td>In-context, non-typical, Black Swan operational images:

1158 (100%)</td></tr>
</table>

	No CuneiForms	No CuneiForms	**CuneiForm Scenarios:** H.36, 37, 38, 39, 40, 41
	Note: The test dataset is identical to the one used in Experiment 3. No pre-processing steps were applied. No augmentations were applied. The training dataset of Exp.4 has a different set of OOC images from the one we used in Exp.1.		
Experiment 5: See link **Total:** 14384 images	**Total:** 9924(69%)	**Total:** 1399(10%)	**Total:** 3061(21%)
	In-context images: 0 **OOC images:** 9924 (100%)	**In-context images:** 0 **OOC images:** 1399 (100%)	**In-context typical operations images**: 3061 (100%)
	No CuneiForms	No CuneiForms	**CuneiForm scenarios**: H.48, 49,50
	Note: The testing dataset is identical to experiment 6. **Applied Pre-processing:** Resize: Stretch to 320x320. No augmentations were applied. No augmentations were applied.		
Experiment 6: See link **Total:** 14479 images	**Total:** 10015 (69%)	**Total:** 1403(10%)	**Total:** 3061(21%)
	In-context typical operations images: 3270 (32%) **OOC situations:** 6745 (67%)	**In-context typical operations images**: 1202 (85%) **OOC situations:** 201 (15%)	**In-context typical operations images**: 3061 (100%)
	CuneiForms scenarios: H.42, 43,44	**CuneiForms scenarios:** H.45, 46,47	**CuneiForms scenarios:** H.48, 49,50
	Applied Pre-processing: Resize: Stretch to 320x320. No augmentations were applied.		
	Total: 10015 (80%)	**Total:** 1405(11%)	**Total:** 1158(9%)

Experiment 7: See link **Total:** 12578 images	**In-context typical operations images**: 3270 (32%) **OOC situations:** 6745 (67%)	**In-context typical operations images**: 1202 (85%) **OOC situations:** 201 (15%)	**In-context, non-typical, Black Swan operational images:** 1158 (100%)
	CuneiForms scenarios: H.42, 43,44	**CuneiForms scenarios:** H.45, 46,47	**Black Swans A CuneiForm Scenarios:** H.36, 37, 38, 39, 40, 41
	Note: The training and validation datasets are identical to those in Experiment 6. The testing dataset is similar to experiments 3 and 4. **Applied Pre-processing:** Resize: Stretch to 320x320. No augmentations were applied.		
Experiment 8.1: See link **Total:** 12561 images	**Total:** 8000 (64%)	**Total:** 1500(12%)	**Total:** 3061(24%)
	In-context non-typical black swan images: 2390 (30%) **OOC images:** 5610 (70%)	**In-context non-typical black swan images:** 701 (47%) **OOC situations:** 799 (63%)	**In-context typical operations images**: 3061 (100%)
	Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41	**Black Swans A CuneiForm Scenarios**: H.36, 37, 38, 39, 40, 41	**CuneiForms scenarios:** H.48, 49,50
	Note: Training and validation are similar to experiment 3. Testing is the same as experiments 6 and 5. **Applied Pre-processing:** Resize: Stretch to 320x320. No augmentations were applied.		
Experiment 8.2: See link **Total:** 10658 images	**Total:** 8000 (75%)	**Total:** 1500 (14%)	**Total:** 1158 (11%)
	In-context non-typical black swan images: 2390 (30%)	**In-context non-typical black swan images:** 701 (47%)	**In-context, non-typical, Black Swan operational images:** 1158 (100%)

	OOC images: 5610 (70%)	**OOC situations:** 799 (63%)	
	Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41	**Black Swans A CuneiForm Scenarios**: H.36, 37, 38, 39, 40, 41	**Black Swans B CuneiForms scenarios:** H.51, 52,53
	Note: Training and validation are similar to experiment 3, 8.1. Testing is on entirely different Black Swans from the ones being trained on.		
	Applied Pre-processing: Resize: Stretch to 320x320. No augmentations were applied.		
Experiment 9.1: See link **Total:** 22374 images	**Total Training:** 14961 (67%)	**Total Valid:** 2903 (13%)	**Total Test:** 4510 (20%)
	In-context non-typical black swan images. and **In-context typical operations images**: 5655 (37%) **OOC images:** 9306 (63%)	**In-context non-typical black swan images.** and **In-context typical operations images**: 1911(67%) **OOC images:** 992 (34%)	**In-context typical operations images**: 3461 (77%) **OOC images:** 1049 (23%)
	Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41	**Black Swans A CuneiForm Scenarios**: H.36, 37, 38, 39, 40, 41	No black swans
	Typical Operations CuneiForm Scenarios: H.42, 43,44	**Typical Operations CuneiForm Scenarios**: H.46,47	**Typical Operations CuneiForm Scenarios**: H.45, 48,49,50
	Applied Pre-processing: Resize: Stretch to 320x320. No augmentations were applied.		
Experiment 9.2:	**Total Training:** 17196 (73%)	**Total Valid:** 5178 (22%)	**Total Test:** 1158 (5%)

<table>
<tr><td rowspan="4">See link

Total: 23532 images</td><td>In-context non-typical black swan images. and In-context typical operations images: 7851 (45%)

OOC images: 9502 (55%)</td><td>In-context non-typical black swan images. and In-context typical operations images: 3165 (61%)

OOC images: 2079 (39%)</td><td>In-context non-typical black swan images: 1158 (100%)</td></tr>
<tr><td>Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td><td>Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td><td>Black Swans B CuneiForm Scenarios: H.51,52,53</td></tr>
<tr><td colspan="2">Typical Operations CuneiForm Scenarios: H.42, 43,44, 45,46,47,48,48,50</td><td>No typical ops images</td></tr>
<tr><td colspan="3">• Applied Pre-processing: Resize: Stretch to 320x320. No augmentations were applied.</td></tr>
<tr><td rowspan="4">Experiment 9.3:

See link

Total: 57924 images</td><td>Total Training: 51588 (89%)</td><td>Total Valid: 5178 (9%)</td><td>Total Test: 1158 (2%)</td></tr>
<tr><td>In-context non-typical black swan images. and In-context typical operations images: 7851 (45%)

OOC images: 9502 (55%)</td><td>In-context non-typical black swan images. and In-context typical operations images: 3165 (61%)

OOC images: 2079 (39%)</td><td>In-context non-typical black swan images: 1158 (100%)</td></tr>
<tr><td>Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td><td>Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td><td>Black Swans B CuneiForm Scenarios: H.51,52,53</td></tr>
<tr><td colspan="2">Typical Operations CuneiForm Scenarios: H.42, 43,44, 45,46,47,48,49,50</td><td>No typical ops images</td></tr>
</table>

<table>
<tr><td></td><td colspan="3">Note: the following pre-processing and augmentations were carried out:

Pre-processing:
• Auto-Orient: Applied
• Resize: Stretch to 320x320
• Auto-Adjust Contrast: Using Adaptive Equalization
Augmentations:
4) Outputs per training example: 3
5) Flip: Horizontal
6) 90° Rotate: Clockwise, Counter-Clockwise
7) Shear: ±15° Horizontal, ±14° Vertical
8) Grayscale: Apply to 15% of images
9) Hue: Between -14° and +14°</td></tr>
<tr><td rowspan="4">Experiment 9.4:

See link

Total: 23533 images</td><td>Total Training: 16788 (71%)</td><td>Total Valid: 4426 (19%)</td><td>Total Test: 2319 (10%)</td></tr>
<tr><td>Black Swan Batch B: 695(4.1%). Black Swan Batch A: 1866 (11.11%). In-context typical operations images: 3273 (19.49%). OOC images: 10954 (65.24%)</td><td>Black Swan Batch B:231(5.22%). Black Swan Batch A:622(14.05%). In-context typical operations images: 1604(36.24). OOC images:1969 (44.49%)</td><td>Black Swan Batch B:232(10%). Black Swan Batch A: 622(26.82%). In-context typical operations images: 1197(51.61%). OOC images: 268 (11.55%)</td></tr>
<tr><td>Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td><td>Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td><td>Black Swans A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td></tr>
<tr><td>Black Swans B CuneiForm Scenarios: H.51,52,53</td><td>Black Swans B CuneiForm Scenarios: H.51,52,53</td><td>Black Swans B CuneiForm Scenarios: H.51,52,53</td></tr>
</table>

<table>
<tr><td></td><td>In-context typical operations images: H.42,43,44</td><td>In-context typical operations images: H.45,46,47</td><td>In-context typical operations images: H.48,49,50</td></tr>
<tr><td rowspan="4">Experiment 10:

See link

Total: 101548 images</td><td>Total Training: 85677 (84%)</td><td>Total Valid: 14713 (14%)</td><td>Total Test: 1158 (1%)</td></tr>
<tr><td>In-context typical operations images:

3270 (4%)

OOC situations:

82257 (96%)</td><td>In-context typical operations images:

1202 (8%)

OOC situations:

13511 (15%)</td><td>In-context, non-typical, Black Swan operational images:

1158 (100%)</td></tr>
<tr><td>CuneiForms scenarios: H.42, 43,44</td><td>CuneiForms scenarios: H.45, 46,47</td><td>Black Swans Batch A CuneiForm Scenarios: H.36, 37, 38, 39, 40, 41</td></tr>
<tr><td colspan="3">Note: this is an extended version of experiment 7. We added a significant number of random, OOC images to the test dataset to assess how increasing the number of images would affect performance compared to Black Swans.

Pre-processing:
• Resize: Stretch to 320x320</td></tr>
</table>