OCTA Defense Research

SPATIAL-OPSEC+INT NG v2.0: A Spatial-AI Testbed for Operational Security and Environmental Intelligence

OCTA Defense Research Laboratory

January 3, 2026

Abstract

We introduce *SPATIAL-OPSEC+INT NG v2.0*, a next-generation spatial–AI testbed for operational security (OPSEC) and environmental intelligence (INT), developed by OCTA Defense Research. The system models environments as multi-layer spatial fields, couples them to multi-modal sensor suites with self-calibrating trust, and drives policy-based deployments of sensing assets and patrol agents. The testbed is designed as a research platform for sensing, situational awareness, and safety—not for weapon control. This document provides a mathematical and algorithmic specification of the grid-based environment, sensor models, semantic fields, policy objectives, and benchmark protocols. It also formalizes the *environment-as-agent* view, where the domain itself exhibits structured, sometimes adversarial, dynamics. The document is intended as a reference for both simulation code and future real-world deployments.

Ethics and Use Disclaimer

This framework is developed exclusively for research in sensing, situational awareness, infrastructure protection, disaster response, environmental monitoring, and OPSEC analysis. It is not designed, intended, or authorized for use in any form of weapon targeting, lethal autonomy, or harm-directing control systems.

Contents

1 Introduction

1.1 Motivation

Operational security and environmental intelligence systems must reason over spatially distributed signals under uncertainty. They combine heterogeneous sensors, partial observability, noisy communications, and evolving threat models. Traditional surveillance frameworks often treat sensing, fusion, and policy as loosely coupled modules; by contrast, *spatial AI* emphasizes end-to-end, field-centric reasoning over continuous space and time.

The **SPATIAL-OPSEC+INT NG** engine is designed as a unified research testbed with the following goals:

- Model environments as dynamic fields rather than static maps.
- Support multi-modal sensing (RGB, thermal, radar, acoustic) with evolving trust.
- Encode high-level OPSEC and INT goals as differentiable objective functionals.
- Enable policy-driven deployment of assets (e.g., cameras, UAVs, patrol units).
- Provide domain presets (open field, urban security, perimeter, critical facility, disaster response, forest border, coastal port, mountain pass, desert test, arctic outpost) as reproducible benchmarks.
- Treat the *environment itself* as an explicit agent, with its own latent state and policy, enabling game-theoretic and robustness analysis.

1.2 Scope and non-goals

The system is explicitly focused on:

- Sensing, situational awareness, environmental coverage.
- Threat *detection* and *localization*, not engagement.
- Research and prototyping of safety-focused spatial AI.

It is *not* intended for:

- Weapon control, targeting loops, or lethal autonomy.
- Real-time command-and-control of kinetic systems.

All algorithms and metrics are designed to emphasize coverage, blind-spot reduction, calibrated uncertainty, and robustness to sensor degradation.

2 Domains, Coherence, and the Structure of Spatial Intelligence

2.1 Domain-first design philosophy

A defining feature of SPATIAL-OPSEC+INT is its *domain-first* architecture. A **domain** is not merely a map; it is the joint specification of:

(a) spatial topology and terrain affordances,

(b) sociotechnical context and operational constraints,

(c) expected traffic priors and anomaly structure,

(d) risk weighting across the environment,

(e) policy envelope (legal, ethical, mission constraints).

In other words, a domain is both a **physical space** and a **theory of how that space tends to behave**.

For example:

- The **OPSEC_OPEN_FIELD** domain assumes weak spatial structure, strong environmental influence, and long-range sensing.
- The **URBAN_SECURITY** domain assumes hard occlusions, structured flow along corridors, and complex social priors.
- The **CRITICAL_FACILITY** domain assumes asymmetric risk: small areas are far more important than others.

This perspective moves beyond raw coverage metrics toward *mission-aligned sensing*: the value of a sensor observation is domain-dependent.

2.2 Coherence as the central organizing principle

Spatial intelligence requires not merely sensing, but *coherent* understanding across space, time, and modalities. We define four interacting coherence classes.

(1) Spatial coherence. Nearby regions tend to have correlated activity unless occlusions or structural discontinuities intervene. This motivates diffusion, smoothing, and spatial priors in the field dynamics.

(2) Temporal coherence. Most environmental processes vary gradually. Temporal coherence supports:

- short-term prediction,
- anomaly definition as deviation from expected temporal evolution,
- filtering and smoothing across time.

(3) Multi-modal coherence. Different modalities (RGB, thermal, radar, acoustic) observe *different projections of the same latent process*. True signal should express *cross-modal compatibility*, while noise and spoofing often fail to do so. This provides a natural basis for trust updating and sensor fault detection.

(4) Policy coherence. A policy should not produce erratic, discontinuous actions in response to small measurement fluctuations. Coherence here means that decision-making is:

- stable,
- explainable,
- robust to noise,
- aligned with domain intent.

Remark 2.1. *Coherence is the* ***antidote to tactical brittleness****. A brittle system reacts; a coherent system understands.*

3 Environment as an Agent

3.1 Motivation for an environment-agent view

Most classical models treat the environment as a passive Markovian background: a transition kernel driven solely by the actions of deployed agents (e.g., sensors, patrol units). For OPSEC and INT, this is insufficient. The *world* itself exhibits:

- structured dynamics (traffic flows, weather, infrastructure),
- adversarial or strategic elements (hostile actors, decoys),
- exogenous shocks (disasters, infrastructure failures),
- slow drifts (urban growth, vegetation, seasonal changes).

We therefore model the environment as an *explicit agent* with a latent state, action space, and policy. This unlocks:

- game-theoretic formulations (red vs blue agents),
- robustness analysis under environment policy shifts,
- domain-specific "personalities" of the world (e.g., calm vs volatile).

3.2 Joint state decomposition

Let X_t denote the full system state at time t. We decompose it as:

$$X_t = (X_t^{\text{env}}, X_t^{\text{blue}}),$$

where:

- X_t^{env} encodes the *environment state*: terrain conditions, weather, infrastructure status, latent threat states, natural flows, crowd dynamics, etc.
- X_t^{blue} encodes the *blue-side state*: sensor positions, trust levels, asset states, fused fields $\mathcal{F}(t)$, and policy internal states.

The environment agent acts through X_t^{env}, while the blue agent (our system) acts through X_t^{blue}.

3.3 Environment action and transition

We posit an environment action space $\mathcal{A}_{\text{env}}$ and a blue action space $\mathcal{A}_{\text{blue}}$. At each time step, the environment chooses an action $u_t^{\text{env}} \in \mathcal{A}_{\text{env}}$ according to a policy π_{env}, while blue chooses $u_t^{\text{blue}} \in \mathcal{A}_{\text{blue}}$ according to policy π_{blue}.

The joint state transition is:

$$X_{t+1} \sim T\big(\cdot \mid X_t, u_t^{\text{env}}, u_t^{\text{blue}}\big),$$

or more explicitly,

$$\begin{aligned} X_{t+1}^{\text{env}} &\sim T_{\text{env}}\big(\cdot \mid X_t^{\text{env}}, X_t^{\text{blue}}, u_t^{\text{env}}\big), \\ X_{t+1}^{\text{blue}} &\sim T_{\text{blue}}\big(\cdot \mid X_t^{\text{blue}}, X_t^{\text{env}}, u_t^{\text{blue}}\big). \end{aligned}$$

This formulation allows the environment to *react* to blue and vice versa.

3.4 Environment policies and domain personalities

Each domain $d \in \mathcal{D}$ corresponds to a family of environment policies $\Pi_{\text{env}}^{(d)}$. For example:

- In **OPSEC_OPEN_FIELD**, π_{env} may generate smooth, weather-driven visibility fluctuations and diffuse crowd movements.
- In **URBAN_SECURITY**, π_{env} may produce peak-hour flows, rush-hour congestion, and night-time lull, with occasional localized surges (events, protests).
- In **CRITICAL_FACILITY**, π_{env} encodes mostly stable conditions with rare, high-impact anomalies representing failures, sabotage attempts, or emergency events.

We can further decompose:

$$\pi_{\text{env}}^{(d)} = \lambda \pi_{\text{natural}}^{(d)} + (1-\lambda)\pi_{\text{adversarial}}^{(d)},$$

where:

- $\pi_{\text{natural}}^{(d)}$ captures benign environmental dynamics,

- $\pi^{(d)}_{\text{adversarial}}$ captures hostile, red-side behaviors,
- $\lambda \in [0, 1]$ controls the mix.

This allows the same domain d to be instantiated as *benign*, *contested*, or *adversarial* by varying λ.

3.5 Game-theoretic view

We can view the overall system as a two-agent partially observable stochastic game (POSG):

- Agent 1 (blue) maximizes $\mathcal{J}_{\text{blue}}$:

$$\mathcal{J}_{\text{blue}}(\pi_{\text{blue}}, \pi_{\text{env}}) = \mathbb{E}\left[\frac{1}{T}\int_0^T J_{\text{ext}}(t)\,\mathrm{d}t\right],$$

  where J_{ext} includes coverage, blind-spot, entropy, trust, cost, and stability terms.

- Agent 2 (environment) may:
  - be indifferent (nature),
  - be adversarial, attempting to maximize OPSEC exposure $E(t)$ and maintain high ambiguity $H(t)$,
  - be cooperative in safety or humanitarian scenarios (e.g., disaster response, where external systems share information).

Adversarial formulations use:

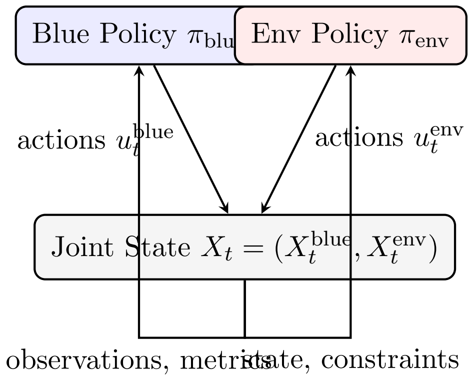
$$\mathcal{J}_{\text{env}} = -\mathcal{J}_{\text{blue}} \quad \text{or} \quad \mathcal{J}_{\text{env}} = \mathbb{E}\left[\frac{1}{T}\int_0^T E(t) + \beta H(t)\,\mathrm{d}t\right].$$

3.6 Environment as an evaluative adversary

Even when the environment is not physically adversarial (no hostile actors), we can choose *virtual adversarial* policies π_{env} to probe blue policy robustness, e.g.:

- worst-case weather patterns relative to sensor placement,
- stress-test patterns of infrastructure failure,
- synthetic waves of decoy signals.

This *environment-as-agent* adversary acts as an automated red-team for spatial-AI policy design.



Blue Policy π_{blu}
Env Policy π_{env}
actions u_t^{blue}
actions u_t^{env}
Joint State $X_t = (X_t^{\text{blue}}, X_t^{\text{env}})$
observations, metrics
state, constraints

Figure 1: Environment-as-agent view: blue and environment policies both act on the joint state, creating a game-theoretic structure for OPSEC and INT.

3.7 Diagram: environment vs blue policy interaction

4 Environment and Spatial Grid

4.1 Discrete grid

We represent the environment as a 2D grid of cells with discrete spatial resolution.

Definition 4.1 (Spatial grid)**.** *Let $n_x, n_y \in \mathbb{N}$ be the grid dimensions. The environment grid is*

$$\mathcal{G} = \{(i,j) \mid i \in \{0,\dots,n_x-1\},\ j \in \{0,\dots,n_y-1\}\}.$$

Each cell (i,j) corresponds to a spatial region of size $\Delta x \times \Delta y$.

We index time in discrete steps $t_k = k\Delta t$ with $\Delta t > 0$ fixed.

4.2 Cell types and terrain masks

Each cell (i,j) has:

- A *terrain type* $\tau_{ij} \in \{\text{free}, \text{building}, \text{forest}, \text{road}, \text{water}, \dots\}$.
- A *static mask* $m_{ij} \in [0,1]$ encoding overall traversability and visibility (e.g., $m_{ij} = 0$ for fully blocked).

We define a terrain mask field $M \in [0,1]^{n_x \times n_y}$,

$$M_{ij} = m_{ij}.$$

4.3 Dynamic entities

We consider three primary categories of entities:

- **Assets** (friendly infrastructure, zones to protect).
- **Threats** (adversarial or unknown agents).
- **Sensors** (fixed or mobile sensing platforms).

We model assets and threats as point entities with positions

$$\mathbf{x}_k(t) \in \mathbb{R}^2, \qquad k \in \mathcal{E}_{\text{asset}} \cup \mathcal{E}_{\text{threat}}.$$

Their motion is updated via discrete dynamics, e.g.,

$$\mathbf{x}_k(t_{n+1}) = \mathbf{x}_k(t_n) + \Delta t \, \mathbf{v}_k(t_n) + \sqrt{\Delta t}\, \Sigma_k^{1/2} \, \eta_n,$$

where $\mathbf{v}_k$ is a velocity, Σ_k is a noise covariance, and η_n is a standard Gaussian random vector. These dynamics are themselves part of the environment-agent evolution T_{env}.

4.4 Figure: grid with assets and sensors

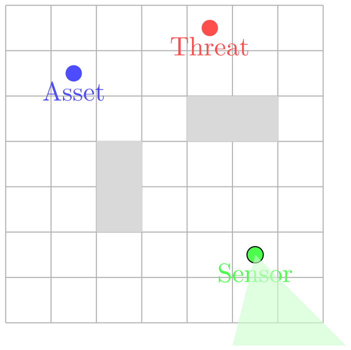


Figure 2: Schematic grid with terrain blocks (gray), asset (blue), sensor (green), and threat (red).

5 Sensor Model and Trust Calibration

5.1 Sensor set and modalities

We define a set of sensors

$$\mathcal{S} = \{1, \dots, S\},$$

where each sensor $s \in \mathcal{S}$ is characterized by:

- A position $\mathbf{x}_s(t) \in \mathbb{R}^2$ (fixed or mobile).
- A modality $m_s \in \{\text{RGB}, \text{THM}, \text{RAD}, \text{AC}\}$: RGB, thermal, radar, or acoustic.
- A field-of-view (FOV) function $\mathsf{FOV}_s(i,j) \in [0,1]$ that encodes relative visibility of grid cell (i,j).
- A trust weight $w_s(t) \in [0,1]$ reflecting reliability.

5.2 Measurement model

At time t, sensor s produces a measurement $y_s(t)$, modeled as

$$y_s(t) = h_s\big(\mathcal{F}(t), \mathbf{x}_s(t)\big) + \epsilon_s(t), \tag{1}$$

where:

- $\mathcal{F}(t)$ is the current semantic field (see Section 6).
- h_s is a modality-specific observation operator.
- $\epsilon_s(t)$ is sensor noise, typically modeled as $\epsilon_s(t) \sim \mathcal{N}(0, \sigma_s^2)$ or more complex.

5.3 Trust update rule

We maintain a trust weight $w_s(t)$ that is updated based on residuals between expected and observed measurements. Let $\hat{y}_s(t)$ be the predicted measurement under the current field estimate. We define the residual

$$r_s(t) = y_s(t) - \hat{y}_s(t),$$

and a trust update,

$$w_s(t + \Delta t) = \mathrm{clip}\Big(w_s(t) - \alpha_s f\big(|r_s(t)|\big) + \beta_s g\big(\mathrm{consistency}_s(t)\big), 0, 1\Big), \tag{2}$$

where:

- $\alpha_s, \beta_s > 0$ are learning rates.
- f penalizes large residuals.
- g rewards consistency with other sensors or prior patterns.
- $\mathrm{clip}(x, 0, 1)$ truncates to $[0, 1]$.

Remark 5.1. *Trust dynamics provide a soft, continuous measure of sensor degradation or failure, which feeds into the policy objective (Section 9) and into OPSEC exposure (Section 10). They also form part of the blue-side state X_t^{blue} in the environment-agent game.*

6 Semantic Field Representation

6.1 Field tensor

We represent spatial semantics as a C-channel tensor

$$\mathcal{F}(t) \in \mathbb{R}_{\geq 0}^{n_x \times n_y \times C},$$

where each channel corresponds to a distinct semantic quantity, e.g.,

$$C = 3, \quad \begin{cases} c = 1: & \text{generic activity intensity,} \\ c = 2: & \text{mobile / moving activity,} \\ c = 3: & \text{periodic / acoustic activity.} \end{cases}$$

We write $\mathcal{F}_{ij}^c(t)$ for the value at cell (i, j) in channel c at time t.

6.2 Field dynamics

A simple evolution model for each channel is:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}_{ij}^c(t) = D_c \Delta_{\mathrm{disc}} \mathcal{F}_{ij}^c(t) - \lambda_c \mathcal{F}_{ij}^c(t) + S_{ij}^c(t) + \xi_{ij}^c(t), \tag{3}$$

where:

- D_c is a diffusion coefficient.
- Δ_{disc} is the discrete Laplacian on the grid.

- λ_c is a decay rate.
- $S_{ij}^c(t)$ is a source term (e.g., detected events, known traffic).
- $\xi_{ij}^c(t)$ is process noise.

The discrete Laplacian can be defined as

$$\Delta_{\text{disc}}\mathcal{F}_{ij}^c = \mathcal{F}_{i+1,j}^c + \mathcal{F}_{i-1,j}^c + \mathcal{F}_{i,j+1}^c + \mathcal{F}_{i,j-1}^c - 4\mathcal{F}_{ij}^c,$$

with appropriate boundary handling.

6.3 Sensor update integration

Measurements $y_s(t)$ induce updates to $\mathcal{F}$ via a Bayesian or heuristic rule. One simple scheme is:

$$\mathcal{F}_{ij}^c(t^+) = \mathcal{F}_{ij}^c(t^-) + \sum_{s\in\mathcal{S}} w_s(t)\,\mathsf{FOV}_s(i,j)\,\gamma_{s,c}\,\phi\big(y_s(t)\big), \tag{4}$$

where:

- t^-, t^+ denote pre- and post-update.
- $\gamma_{s,c}$ is a modality-to-channel coupling weight.
- ϕ is a non-linear mapping from measurement to field increment.

6.4 Figure: semantic field example

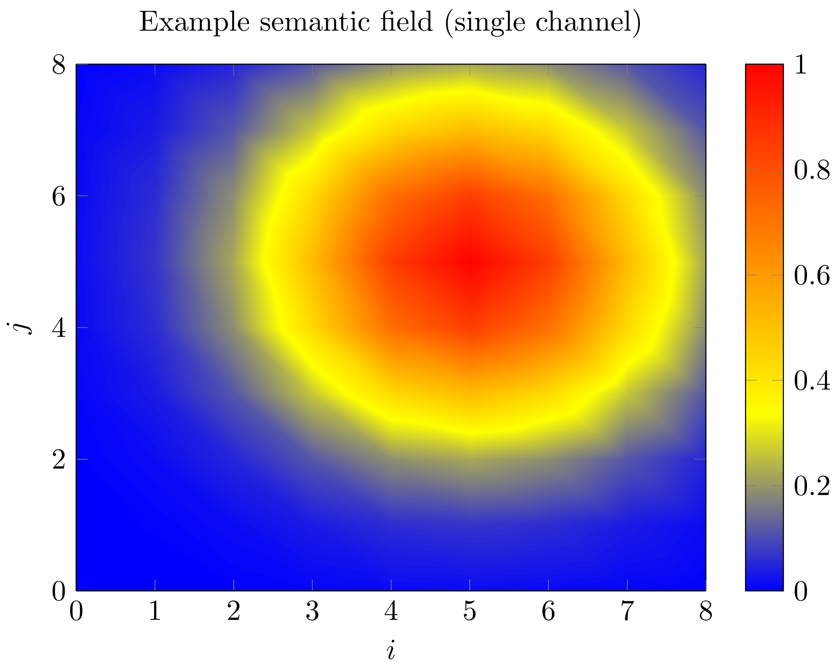


Figure 3: Synthetic semantic field showing a central hotspot.

7 From Sensing to Intelligence

Operational intelligence matures through several layers. We formalize this lifecycle below, linking raw spatial measurements to structured INT.

7.1 Layer 1: Physical sensing

Raw measurements are physical signals:

$$y_s(t) \in \mathbb{R}^k,$$

corrupted by noise, occlusion, and adversarial effects. They are bounded in scope (field-of-view), quality (SNR), and trust.

7.2 Layer 2: Perceptual fields

Measurements update the semantic field $\mathcal{F}(t)$, which encodes:

- intensity of activity,
- mobility likelihood,
- periodic or acoustic signatures,
- threat priors (via source terms $S^c_{ij}(t)$).

This stage transforms raw sensor packets into a consistent, grid-aligned perceptual representation.

7.3 Layer 3: Belief fields

We maintain a normalized belief field

$$P_{ij}(t),$$

which estimates the probability of relevant threat presence. This field is influenced not only by sensing, but also by:

- motion models for threats and assets,
- sociotechnical priors (e.g., likely routes),
- domain constraints and risk weights,
- environment-agent actions via T_{env}.

7.4 Layer 4: INT synthesis

Intelligence synthesis emerges when:

(a) beliefs become structured (non-uniform and interpretable),

(b) uncertainty decreases meaningfully in critical areas,

(c) coherence persists across time and sensor modes.

At this layer, the system supports decision-grade situational awareness.

Definition 7.1 (INT maturity)**.** *We define INT maturity at time t as the functional*

$$M(t) = \frac{C(t)}{H(t) + \epsilon},$$

which measures the ratio of spatial certainty to residual ambiguity.

High $M(t)$ means the system "knows what it knows" and is not merely collecting data.

8 Uncertainty as a Field

Uncertainty is not a scalar — it is a **geometry** over the environment. Blind-spots, partial visibility, and noisy modalities create non-uniform structures in the uncertainty field.

We define local uncertainty:

$$U_{ij}(t) = -P_{ij}(t) \log(P_{ij}(t) + \varepsilon),$$

and global uncertainty:

$$H(t) = \sum_{(i,j)} U_{ij}(t).$$

This field evolves as:

- trust collapses on degraded sensors,
- occlusions shift (e.g., dynamic obstacles, weather),
- policies redirect focus and redistribute coverage,
- environment-agent actions steer traffic and threat behavior.

Good policy does not merely reduce $H(t)$ — it *shapes it*, preventing dangerous uncertainty concentration near critical assets.

9 OPSEC and INT Objective Functional

9.1 Coverage and blind-spot metrics

Let Ω denote the set of all grid cells. At time t, define a coverage score $C(t)$ as

$$C(t) = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \Psi_{ij}(t), \tag{5}$$

where $\Psi_{ij}(t) \in [0,1]$ encodes effective coverage of cell (i,j) given sensor FOV and trust:

$$\Psi_{ij}(t) = 1 - \prod_{s\in\mathcal{S}} \big(1 - w_s(t)\, \mathsf{FOV}_s(i,j)\big). \tag{6}$$

A blind-spot penalty is then

$$B(t) = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \big(1 - \Psi_{ij}(t)\big)\, \chi_{ij}^{\text{critical}}, \tag{7}$$

where $\chi_{ij}^{\text{critical}} \in \{0,1\}$ indicates critical regions (e.g., near assets, chokepoints).

9.2 Uncertainty and entropy

We maintain a probabilistic belief over threat presence at each cell, $P_{ij}(t) \in [0,1]$, with $\sum_{(i,j)} P_{ij}(t) = 1$. A global entropy measure is

$$H(t) = -\sum_{(i,j)\in\Omega} P_{ij}(t) \log\bigl(P_{ij}(t) + \varepsilon\bigr), \tag{8}$$

where $\varepsilon > 0$ prevents singularities.

Entropy quantifies spatial uncertainty about threat location; OPSEC policies often aim to reduce $H(t)$ in critical zones, yielding higher INT maturity $M(t)$.

9.3 Sensor cost

Let $c_s(a_s(t))$ be the operational cost of the chosen action $a_s(t)$ for sensor s (e.g., moving a UAV, switching modes). The total cost rate is

$$K(t) = \sum_{s\in\mathcal{S}} c_s\bigl(a_s(t)\bigr). \tag{9}$$

9.4 Trust and consistency

We define a trust penalty

$$L_{\text{trust}}(t) = \frac{1}{S} \sum_{s\in\mathcal{S}} \bigl(1 - w_s(t)\bigr),$$

which measures average degradation. More refined variants can penalize disagreement between sensors with overlapping FOV.

9.5 Composite objective

We define a composite instantaneous objective functional

$$J(t) = w_{\text{cov}}\, C(t) - w_{\text{blind}}\, B(t) - w_{\text{ent}}\, H(t) - w_{\text{cost}}\, K(t) - w_{\text{trust}}\, L_{\text{trust}}(t), \tag{10}$$

where:

- $w_{\text{cov}}, w_{\text{blind}}, w_{\text{ent}}, w_{\text{cost}}, w_{\text{trust}} \geq 0$ are scalar weights.

For an episode over time horizon $[0,T]$, the objective of a policy π is

$$\mathcal{J}(\pi) = \frac{1}{T} \int_0^T J_\pi(t)\, \mathrm{d}t, \tag{11}$$

where $J_\pi(t)$ is given by (10) under the actions chosen by policy π.

10 OPSEC Framing

10.1 Exposure vs protection

OPSEC analysis asks:

How exposed are we if an adversary observes or exploits gaps?

Blind–spot structure and sensor trust degradation map directly to exposure risk. We define OPSEC exposure:

$$E(t) = \sum_{(i,j)} \chi_{ij}^{\text{critical}} \big(1 - \Psi_{ij}(t)\big).$$

This is the part of the blind-spot penalty *that matters most.* While $B(t)$ averages over the grid, $E(t)$ isolates the risk in mission-critical regions.

10.2 Deception and decoys

Adversaries may induce false hotspots (e.g., heat signatures or acoustic decoys). Multi-modal coherence and trust dynamics are the countermeasure:

$$\text{decoys} \;\Rightarrow\; \text{low cross-modal alignment.}$$

Inconsistency between modalities (e.g., strong thermal response without supporting acoustic or radar evidence) can be treated as a hypothesis for deception, driving local trust reduction.

10.3 Resilience

A resilient OPSEC system:

- maintains bounded exposure despite failures,
- reallocates attention coherently,
- preserves mission priorities under resource constraints.

Policy brittleness is itself an OPSEC risk; this motivates explicit penalties for unstable decision behavior (Section 12).

11 Policy Families and Deployment Logic

We consider a family of policies $\mathcal{P}$ operating on state summaries built from $\mathcal{F}(t)$, sensor states, and domain context.

11.1 State abstraction

Let $z(t)$ denote a compact state summary at time t, aggregating:

- field statistics (e.g., hotspots, gradients, entropy),
- sensor positions and trust levels,
- asset and threat belief distributions,
- domain tag (e.g., OPSEC_OPEN_FIELD, URBAN_SECURITY),
- features derived from environment-agent state X_t^{env}.

Formally,

$$z(t) = \Phi\big(\mathcal{F}(t), \{\mathbf{x}_s(t), w_s(t)\}_{s\in\mathcal{S}}, X_t^{\text{env}}, \text{context}\big).$$

A policy π is a mapping

$$\pi : z(t) \mapsto a(t) = \{a_s(t)\}_{s\in\mathcal{S}},$$

where $a_s(t) \in \mathcal{A}_s$ is the action for sensor s.

11.2 Policy archetypes

We define four archetypal policies:

Coverage policy ($\pi_{\mathbf{cov}}$). Maximizes spatial coverage with soft preference for critical regions:

$$\pi_{\text{cov}}(z) = \arg\max_a C(z,a) - \lambda K(z,a).$$

Hotspot policy ($\pi_{\mathbf{hot}}$). Focuses assets around high-activity or high-uncertainty hotspots:

$$\pi_{\text{hot}}(z) = \arg\max_a \sum_{(i,j)\in\text{hotspots}(z)} \Psi_{ij}(z,a) - \lambda K(z,a).$$

Perimeter policy ($\pi_{\mathbf{perim}}$). Allocates sensors along boundaries and chokepoints:

$$\pi_{\text{perim}}(z) = \arg\max_a C_{\text{perim}}(z,a) - \lambda K(z,a),$$

where C_{perim} is coverage restricted to boundary cells.

Hybrid policy ($\pi_{\mathbf{hybrid}}$). Blends the others using a context-dependent mixture:

$$\pi_{\text{hybrid}}(z) = \sum_k \alpha_k(z)\,\pi_k(z), \quad \sum_k \alpha_k(z) = 1,\ \alpha_k(z) \geq 0, \tag{12}$$

with $k \in \{\text{cov}, \text{hot}, \text{perim}\}$.

12 Decision-Stability and Policy Geometry

Let the policy be π. Policy stability is measured by sensitivity:

$$S_\pi = \mathbb{E}\,\|\pi(z+\delta z) - \pi(z)\|,$$

for small perturbations δz in the state summary.

Low S_π corresponds to stable behavior; high S_π indicates brittleness, where small observational noise can cause large shifts in action.

We explicitly penalize instability via

$$L_{\text{stab}} = \lambda_{\text{stab}} S_\pi.$$

An extended objective therefore may include:

$$J_{\text{ext}}(t) = J(t) - w_{\text{stab}} L_{\text{stab}}(t),$$

so that the policy learns not only *what to do*, but *how smoothly to change its mind*.

This defines a geometry on policy space where desirable policies lie in low-curvature regions (with respect to perturbations in z), yielding robust, interpretable behavior.

13 Simulation Engine and Algorithms

13.1 Episode loop

A simulation episode is a sequence of time steps $t_0, \dots, t_N$. At each step, we:

1. Environment-agent chooses u_t^{env} from π_{env}.
2. Blue-side policy chooses u_t^{blue} from π_{blue}.
3. Evolve entities (assets, threats) via T_{env}.
4. Evolve fields via (3).
5. Generate sensor measurements via (1).
6. Update trust via (2).
7. Fuse measurements into fields via (4).
8. Compute state summary $z(t)$.
9. Log metrics for benchmarking.

13.2 Algorithm: single episode

Algorithm 1 Single SPATIAL-OPSEC+INT episode with environment agent

```
Require: initial state $X_0^{\text{env}}, X_0^{\text{blue}}$, policy $\pi_{\text{blue}}$, environment policy $\pi_{\text{env}}$, horizon $T$, step $\Delta t$
 1: $t \leftarrow 0$
 2: while $t < T$ do
 3:     $u_t^{\text{env}} \leftarrow \pi_{\text{env}}(X_t^{\text{env}}, X_t^{\text{blue}})$
 4:     $u_t^{\text{blue}} \leftarrow \pi_{\text{blue}}(X_t^{\text{blue}}, X_t^{\text{env}})$
 5:     sample $X_{t+\Delta t}^{\text{env}}, X_{t+\Delta t}^{\text{blue}}$ via transition $T$
 6:     Evolve $\mathcal{F}(t)$ via field dynamics (3)
 7:     for each sensor $s \in \mathcal{S}$ do
 8:         sample measurement $y_s(t)$ via (1)
 9:         update trust $w_s(t + \Delta t)$ via (2)
10:     end for
11:     update field $\mathcal{F}(t^+)$ with measurements via (4)
12:     compute coverage $C(t)$, blind-spot $B(t)$, entropy $H(t)$, cost $K(t)$
13:     form summary state $z(t)$
14:     log metrics and state
15:     $t \leftarrow t + \Delta t$
16: end while
```

13.3 Benchmark grid and automated runs

For systematic evaluation, we define a set of benchmark scenarios:

$\mathcal{D} = \{\text{OPSEC\_OPEN\_FIELD}, \text{URBAN\_SECURITY}, \text{PERIMETER\_MONITOR}, \text{CRITICAL\_FACILITY}, \text{DISAST}$

For each domain $d \in \mathcal{D}$, we define:

- Terrain mask $M^{(d)}$.
- Asset layout and threat priors.
- Sensor layouts and modalities.
- Environment policy family $\Pi_{\text{env}}^{(d)}$.

We can then perform automated sweeps over:

- Blue policies $\pi_{\text{blue}} \in \mathcal{P}$.
- Environment policies $\pi_{\text{env}} \in \Pi_{\text{env}}^{(d)}$.
- Objective weights $w_\cdot$.
- Sensor budgets and trust dynamics.

Algorithm 2 Benchmark sweep over domains, blue policies, and environment policies

```
Require: domain set $\mathcal{D}$, blue policy set $\mathcal{P}$, environment policy sets $\Pi_{\text{env}}^{(d)}$, config grid $\Xi$
 1: for each domain $d \in \mathcal{D}$ do
 2:     for each blue policy $\pi_{\text{blue}} \in \mathcal{P}$ do
 3:         for each environment policy $\pi_{\text{env}} \in \Pi_{\text{env}}^{(d)}$ do
 4:             for each config $\xi \in \Xi$ do
 5:                 sample random seed $\omega$
 6:                 run episode with domain $d$, $\pi_{\text{blue}}$, $\pi_{\text{env}}$, config $\xi$, seed $\omega$
 7:                 store metrics $\mathcal{M}(d, \pi_{\text{blue}}, \pi_{\text{env}}, \xi, \omega)$
 8:             end for
 9:         end for
10:     end for
```

14 Metrics and Visualization

14.1 Key metrics

For each run, we track:

- Time-averaged coverage $\bar{C}$.
- Time-averaged blind-spot penalty $\bar{B}$.
- Time-averaged entropy $\bar{H}$.
- Time-averaged cost $\bar{K}$.
- Mean trust level $\bar{w}$ and trust variance.
- Detection latency and false-alarm rates.
- Exposure $\bar{E}$ under environment policies.

The core performance measure can be the time-averaged objective

$$\overline{\mathcal{J}} = \frac{1}{N} \sum_{k=0}^{N-1} J(t_k),$$

where $t_k = k\Delta t$.

14.2 Example time-series plot

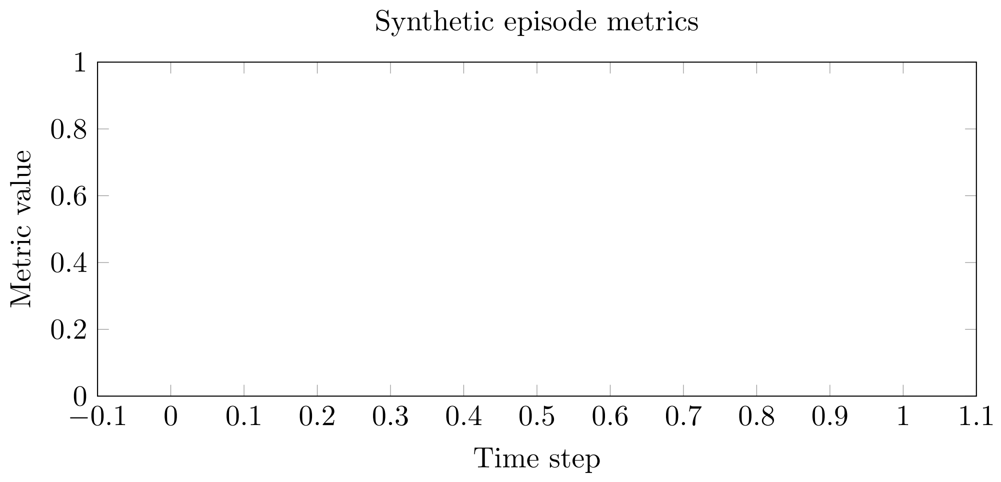


Figure 4: Synthetic episode showing coverage increasing while blind-spots shrink.

15 Empirical Illustration: Urban Crowd Monitoring Log

15.1 URBAN-crowd_monitoring domain

One of the terrain presets in SPATIAL-OPSEC+INT is the `URBAN-crowd_monitoring` domain. It instantiates:

- a dense grid with building occlusions and streets,
- a ground-truth crowd activity field (moving clusters),
- a fleet of heterogeneous sensors (fixed and mobile),
- an OPSEC policy tuned for civilian crowd safety and anomaly detection.

Figure 5 shows a final-time snapshot from a typical run in this domain.

Ground truth vs belief. The left panel is the "god's-eye" activity field $A_{ij}(T) \in [0, 1]$ at the end of an episode (crowd intensity). The middle panel is the inferred belief $P_{ij}(T)$ over activity, computed by the fusion pipeline. Visually, the system has learned a sparse, high-confidence estimate around the main clusters, with low probability mass elsewhere.

The quality of this match is captured by the *environment correlation* metric defined in Section 16.

urban_crowd_log.png

Figure 5: Example log from `URBAN-crowd_monitoring`. Left: ground-truth crowd activity field $A_{ij}(T)$. Middle: inferred belief field $P_{ij}(T)$ over activity. Right: assets and sensors overlaid on the belief field, with distinct markers/colors for different asset types.

Assets on grid. The right panel overlays

- the belief field $P_{ij}(T)$ (yellow/white = higher belief),
- the positions of fixed sensors, mobile units, and protected assets, each rendered with a marker and color.

In the `URBAN-crowd_monitoring` domain, the policy attempts to arrange these assets so that high-belief regions are well covered, while also maintaining perimeter awareness and avoiding blind-spots near critical infrastructure.

16 Per-domain Terrain Mapping and Metrics

16.1 Terrain / policy combinations

The benchmark suite spans multiple terrain–policy combinations, such as:

- `OPSEC-global_coverage`,
- `OPSEC-hotspot_tracking`,
- `OPSEC-balanced`,
- `URBAN-infrastructure_monitoring`,
- `URBAN-crowd_monitoring`,
- `URBAN-hotspot_focus`,
- `PERIMETER-ring_coverage`,
- `PERIMETER-breach_focus`,
- `CRITICAL-dense_coverage`,
- `CRITICAL-fallback`,
- `DISASTER-urban_search`,
- `DISASTER-perimeter_triage`,
- `FOREST-border_watch`,
- `FOREST-deep_patrol`,
- `COASTAL-harbor_coverage`,
- `COASTAL-channel_focus`,
- `MOUNTAIN-pass_watch`,
- `MOUNTAIN-valley_focus`.

Each label implicitly encodes:

$$(\text{terrain}, \pi^{(d)}_{\text{env}}, \pi^{(d)}_{\text{blue}}),$$

where terrain and the two policies give that domain its "personality."

16.2 Environment correlation metric

For a given domain and run, let $A_{ij}(T)$ denote the final ground-truth activity field (e.g., crowd intensity) and $P_{ij}(T)$ the final belief field. We define

$$\text{env\_corr\_final} = \text{corr}(\text{vec}(A(T)),\ \text{vec}(P(T)))\,,$$

where vec flattens a field into a vector and corr is the Pearson correlation coefficient.

Values near 1 indicate that high-belief regions coincide with true activity hotspots; low or negative values indicate mismatch or mislocalization.

16.3 Final coverage metric

Recall $\Psi_{ij}(t)$ from (6). The per-domain coverage at final time is

$$\text{coverage\_final} = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \Psi_{ij}(T),$$

which measures how much of the grid is effectively "seen" by the sensor ensemble at episode end.

In domains such as `OPSEC-global_coverage`, the policy directly optimizes this quantity; in domains such as `URBAN-crowd_monitoring` or `CRITICAL-fallback`, the policy may trade some global coverage for tight focus around critical regions or high-risk flows.

16.4 Per-domain metric plots

Figure 6 shows bar plots of env_corr_final (top) and coverage_final (bottom) across the terrain/policy combinations above.

Typical patterns:

- Highly structured, localized domains (e.g., `CRITICAL-dense_coverage`, `URBAN-hotspot_focus`) can achieve high `env_corr_final` even with modest global coverage.
- Broad-coverage domains (e.g. `OPSEC-global_coverage`, `FOREST-border_watch`) show higher `coverage_final` but slightly lower correlation, reflecting trade-offs between "see everywhere" and "fit details perfectly."
- Disaster and fallback modes often exhibit uneven coverage, but prioritize rapid uncertainty reduction in specific regions.

From the environment-as-agent perspective, these bar-plots act as *fingerprints* of how blue policy interacts with a given environment personality.

17 Digital Twin Hooks and Scenario Cloning

17.1 Digital twin abstraction

The engine supports a digital-twin mindset: a single configuration can be instantiated in multiple *clones* representing different hypotheses or interventions.

Let Θ denote a configuration (terrain, assets, sensors, priors). A digital twin instance is an indexed copy (Θ, ω) where ω encodes:

- Random seed.
- Intervention scenario (e.g., sensor failure, weather condition, adversarial environment policy).
- Policy modifications.

17.2 What-if cloning

To evaluate robustness, we can run:

$$\{\text{episode}(\Theta, \omega_k)\}_{k=1}^{K},$$

with controlled variations (e.g., knocking out a subset of sensors, escalating adversarial behavior in π_{env}). The resulting distribution over metrics quantifies sensitivity and robustness.

per_domain_metrics.png

Figure 6: Per-domain performance metrics. Top: `env_corr_final` — correlation between inferred belief and ground-truth activity at episode end. Bottom: `coverage_final` — mean effective coverage over the grid at episode end. Each bar corresponds to a specific terrain/policy configuration.

17.3 OPSEC-focused experiments

Examples:

- Simulate targeted sensor tampering (trust collapse on a subset of sensors).
- Simulate introduction of adversarial decoys (false hotspots).
- Evaluate how quickly policies reconfigure to maintain coverage and keep $B(t)$ and $E(t)$ bounded under such disruptions.
- Stress-test blue policies against worst-case environment agents in $\Pi^{(d)}_{\text{env}}$.

18 Conclusion and Extensions

We have defined a spatial-AI testbed for OPSEC and environmental intelligence that unifies:

- Grid-based environments with semantic field dynamics.
- Multi-modal sensor models with trust calibration.
- OPSEC- and INT-specific objective functionals, including exposure and uncertainty geometry.
- Policy archetypes, decision-stability penalties, and benchmark protocols across diverse domains.
- An explicit *environment-as-agent* formulation, enabling game-theoretic and robustness analysis.

Future extensions include:

- Learning-based policies (e.g., deep RL over $z(t)$) tuned on $\mathcal{J}$ and tested against increasingly capable environment agents.
- More realistic sensor physics and atmospheric effects.
- Multi-layer 3D environments (multi-floor urban, underground structures).
- Integration with real-world sensor logs and geospatial data.
- Systematic study of deception, decoy strategies, and countermeasures, using π_{env} as an automated red team.

The present specification is aligned with the SPATIAL-OPSEC+INT NG v2.0 engine implementation and is intended to serve as a stable foundation for further research and development within OCTA Defense Research.