

Verified Outcome Gated Contextual Plasticity for Safe Adaptive Decision Systems

A Domain Extensible Graph Architecture with Context Isolation and Execution Faithful Explanations

Sara Spencer

CogniMind AI, LLC

Patent status: U.S. provisional patent application filed September 19, 2026. Patent pending.

Public Edition v1.0 | Published September 21, 2026 | CogniMind AI, LLC

Publication notice: This paper describes a patent-pending architecture and a testable evaluation protocol. Controlled implementation testing is underway. The paper does not report completed empirical trials, validated performance claims, production coefficients, calibration data, customer-specific mappings, or other confidential implementation parameters.

Abstract

Adaptive decision systems must learn from experience without allowing frequent use, unverified feedback, or context mismatch to harden unsafe behavior. This paper introduces verified-outcome-gated contextual plasticity, a domain-extensible graph architecture in which connection strength and connection lifecycle state are distinct. Typed graph paths represent events, context, constraints, decisions, actions or communications, and outcomes. A connection may be reinforced numerically through qualified evidence, but promotion from temporary to durable status additionally requires explicit state-transition guards. Learning is isolated by canonical context keys, action authorization is separated from learning authorization, and outcome evidence is screened for provenance, integrity, temporal ordering, resource identity, independence, and causal attribution. Protected contextual influence matrices provide bounded effects while execution metadata yields factor, path, contrastive, and counterfactual explanations without disclosing proprietary coefficients. The architecture records immutable versions of events, contexts, graphs, policies, matrices, actions, outcomes, and learning transitions to support reconstruction and rollback. Worked design cases in governance and cybersecurity, autonomous flight, and resilient communications illustrate how a common learning lifecycle can coexist with domain-specific safety rules. We specify falsifiable hypotheses and an evaluation protocol for reliability, transfer containment, explanation fidelity, and recovery. The contribution is an architectural separation of use, verified learning, state promotion, action authority, and explanation that supports adaptive behavior in safety- and governance-sensitive environments.

Keywords: adaptive graphs; contextual plasticity; verified outcomes; explainable artificial intelligence; safe learning; continual learning; autonomous systems; governance risk and compliance

1 Introduction

Machine-learning systems often collapse several different propositions into one weight update: an action was selected, the action occurred, a favorable observation followed, and the model should generalize from the observation. Those propositions are not equivalent. Selection frequency can reflect policy defaults, feedback loops, or limited alternatives rather than utility. An observed result can be stale, manipulated, produced by another intervention, or valid only for one configuration. A useful adaptive system therefore needs explicit control over what was used, what was verified, what may be learned, where the learning applies, and when a learned relationship becomes operationally trusted.

The motivating analogy combines stigmergic search and biological plasticity. Ant-colony optimization reinforces successful routes while evaporation weakens unused ones [1]. Synaptic and continual-learning

models distinguish rapid adaptation from slower consolidation [2-6]. These mechanisms inspire useful computational patterns, but they do not by themselves supply evidence qualification, operational authority, context isolation, deterministic reconstruction, or domain-specific safety enforcement. Likewise, knowledge-graph reinforcement learning can discover multi-hop paths [7,8], but a rewarded path is not automatically an authorized action or a durable rule.

We propose verified-outcome-gated contextual plasticity (VOCP), an architecture for graph-based adaptive decision systems. VOCP separates five decisions: path eligibility, path ranking, action authorization, outcome verification, and learning authorization. It further separates a connection numerical value from its lifecycle state. Repeated traversal cannot promote a connection unless evidence and transition policies are satisfied. The design is intended for systems where explanations, reversibility, or safe bounded autonomy matter, including governance, risk and compliance (GRC), cybersecurity, communications, robotics, and autonomous vehicles.

This paper makes four contributions:

1. A typed graph and context-key formalism that prevents learned relationships from silently transferring across incompatible operating conditions.
2. A guarded connection lifecycle in which verified outcome evidence, not repetition alone, controls operational promotion, demotion, quarantine, and revocation.
3. A dual-gate control model that separates permission to execute an action from permission to update the learning state.
4. An execution-faithful explanation and replay model that exposes evidence, factor directions, exclusions, constraints, counterfactuals, and version identifiers while protecting proprietary influence-matrix entries.

2 Related Work

2.1 Stigmergic Search and Plasticity

Ant Colony System uses distributed path construction, pheromone reinforcement, and evaporation to balance exploitation and exploration [1]. Hebbian learning and complementary learning systems motivate activity-dependent change and the coexistence of rapid episodic adaptation with slower consolidation [2,3]. Elastic weight consolidation protects parameters important to earlier tasks [4], while differentiable plasticity and dynamic continual-learning architectures learn or allocate plastic components [5,6]. VOCP differs by treating operational trust as an explicit, auditable state transition governed by evidence and context, rather than as a direct consequence of weight magnitude or use frequency.

2.2 Graph Reasoning and Reinforcement Learning

DeepPath and MINERVA cast multi-hop knowledge-graph reasoning as sequential decision problems [7,8]. Their central objective is reasoning accuracy or reward. VOCP uses graph sequences for operational decisions but introduces eligibility filtering before ranking, domain policy gates after ranking, and a verified outcome object before learning. The architecture can host reinforcement learning, probabilistic search, symbolic rules, or deterministic ranking without equating any one optimizer with the governance mechanism.

2.3 Safe and Assured Learning

Constrained policy optimization seeks reward improvement subject to constraints [9], and shielding blocks unsafe actions using formal safety knowledge [10]. Runtime-assurance and Simplex-style architectures place a verified safety mechanism around a complex controller [11]. VOCP is compatible with these approaches but adds a second governance boundary: an action can be safe and authorized while its outcome remains ineligible to train the production graph. Conversely, simulation may be permitted to update only a shadow-scoped connection.

2.4 Explainable Artificial Intelligence

LIME, SHAP, counterfactual explanations, and interpretable-model research provide influential approaches to explaining predictions [12-15]. VOCP does not prescribe a single attribution method. It requires that every explanation bind to the actual event, context, path-ranking metadata, policy results, and methodology versions used at execution. This permits parameter secrecy while enabling fidelity tests: disclosed directions, constraints, and exclusions can be compared with the recorded computation.

3 Problem Formulation

At decision time t , let the versioned adaptive graph be $G_t = (V_t, E_t)$. Nodes have types drawn from a domain-package schema. A connection e in E_t has numerical strength se , uncertainty ue , lifecycle state ze , context key ke , policy version pe , and counters derived only from accepted outcome records. Let ct be the canonical context key for an event. Eligibility is evaluated before ranking:

$$I(e, ct) = 1 \text{ iff } contextMatch(ke, ct) \text{ and } lifecycleEligible(ze) \text{ and } evidenceEligible(e) \text{ and } policyEligible(e).$$

A candidate path P is a typed sequence whose connections all satisfy $I(e, ct) = 1$. Its ranking score may combine learned strength, contextual relevance, expected utility, uncertainty, information value, and execution cost:

$$J(P | ct) = \text{sum over } e \text{ in } P \text{ of } [w1 \ se + w2 \ relevance(e, ct) + w3 \ utility(e, ct) - w4 \ ue - w5 \ cost(e, ct)].$$

The expression is illustrative; VOCP does not depend on a linear score. The operational requirement is that the system record the factor values and methodology version used, and that an ineligible connection cannot win by having a larger strength.

An action a selected from P is executed only if the action gate $A(a, ct, policy)$ permits it. An outcome record ot is accepted for learning only if the verifier $V(ot)$ assigns a policy-qualified disposition and the separate learning gate $L(ot, ct, policy)$ permits the update. A proposed transition $T(ze \text{ to } ze\text{-prime})$ is committed only when all transition guards hold and the connection version has not changed since the proposal was computed.

4 Architecture

Verified outcome gated adaptive graph architecture

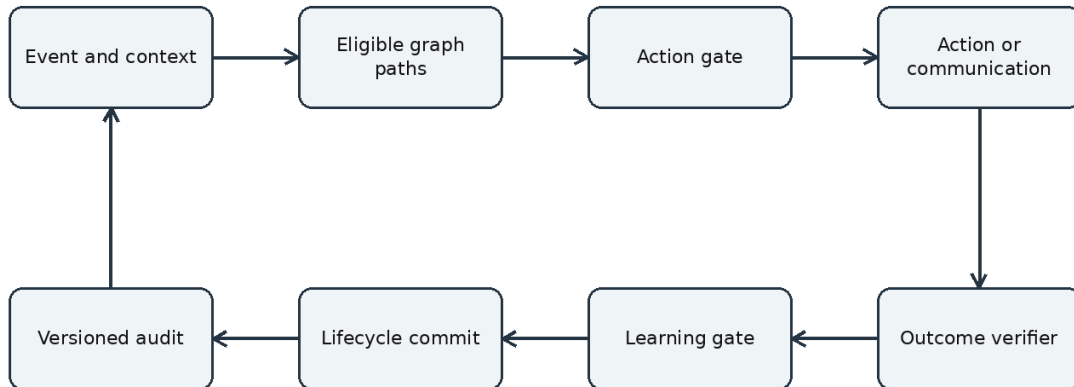


Figure 1. VOCP separates path eligibility and ranking, action authorization, outcome verification, learning authorization, lifecycle transition, and immutable audit.

4.1 Typed Graph and Domain Package

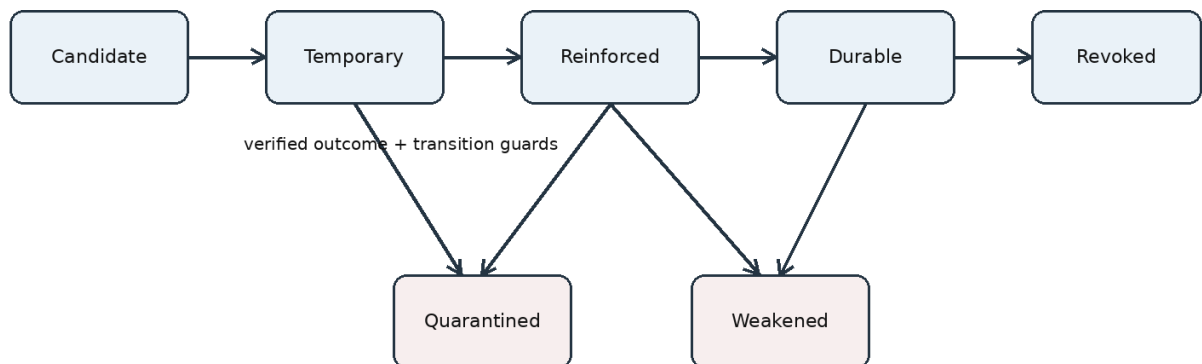
A domain package defines node and connection types, typed ports, compatibility predicates, required context dimensions, action classes, constraints, outcome schemas, verification rules, safe-state behavior, simulation adapters, and explanation templates. The common engine supplies lifecycle, versioning, audit, and replay services. A package that defines consequential actions but omits verification, rollback, or safe-state interfaces is rejected.

4.2 Canonical Context Keys

A context key is a versioned representation of the conditions under which a connection may be selected or updated. Dimensions may identify tenant, system boundary, asset or vehicle class, hardware and software configuration, topology, geography, weather, mission, framework revision, threat condition, evidence source, or control implementation. Exact, hierarchical, and explicitly authorized wildcard matches are permitted. Generalization creates a new scoped connection in a non-durable state; it does not broaden an existing key in place.

4.3 Guarded Connection Lifecycle

Connection lifecycle with guarded transitions



Traversal or repetition alone cannot promote a connection. State and strength are separate variables.

Figure 2. Lifecycle transitions are explicit policy events. Numerical reinforcement cannot bypass verification or transition guards.

Each connection records lifecycle state separately from strength. Candidate connections have not passed structural validation. Temporary connections can participate in simulation or policy-approved operation. Reinforced and durable states require progressively stronger evidence, diversity, time, and approval conditions. Contradictory or causally ambiguous evidence can quarantine a connection; decay can weaken it; verified failure or policy change can revoke it. Every proposal, rejection, and commit is recorded as a transition event.

4.4 Outcome Verification and Causal Attribution

An action record specifies the expected result, observation interval, affected resource, selected path, graph snapshot, and verification policy. Outcome evidence is evaluated for provenance, integrity, source reliability,

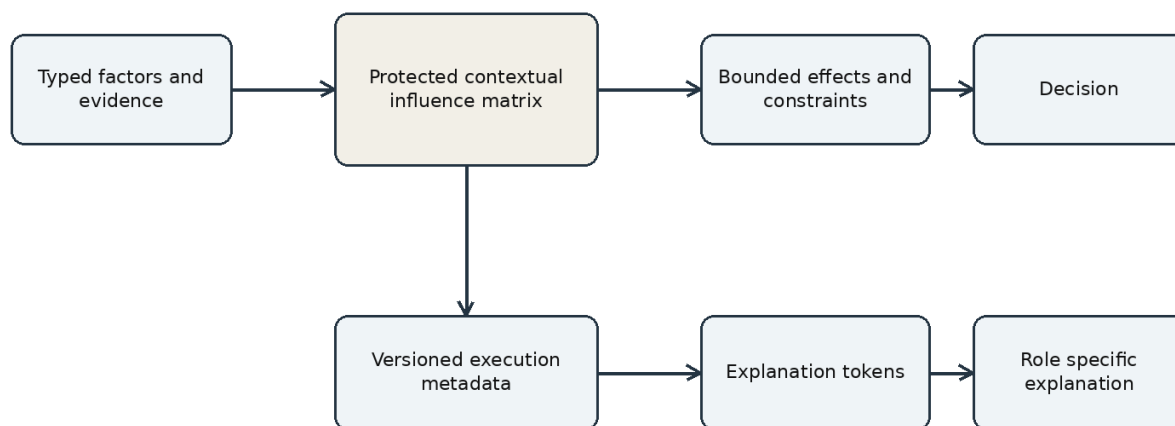
resource identity, temporal order, independence, persistence, and causal attribution. Where several actions plausibly caused an outcome, the verifier may allocate bounded credit, quarantine the update, or require a controlled comparison. Action-generated evidence is not automatically independent verification.

4.5 Dual Authorization

The action gate and learning gate may reach different decisions. Emergency action can be authorized while learning is frozen. Simulation can update only a shadow context. A safe but poorly attributable outcome can be recorded without changing the graph. This separation makes operational permissibility independent from epistemic confidence.

5 Protected Influence Matrices and Explainability

Protected influence matrix and execution faithful explanation



Exact coefficients remain protected; factor direction, evidence, exclusions, constraints, and version identifiers remain

Figure 3. Protected influence matrices return bounded effects and explanation tokens. The explanation remains tied to actual execution without revealing exact entries.

A protected contextual influence matrix transforms typed factors into bounded score, confidence, utility, authorization, or action-selection effects. Matrix denotes a sparse matrix, tensor, multidimensional lookup structure, hierarchical table, graph-indexed parameters, rule-linked coefficients, learned parameters, or a composition. Coordinates may depend on node type, domain, tenant, context, evidence class, control class, action class, outcome class, or methodology version. Constraint entries can override numerical effects.

The matrix executes in an isolated methodology service. The service receives factor identifiers and bounded values and returns effects, constraint results, confidence contributions, and explanation tokens. A signed methodology manifest binds graph, package, matrix, and policy versions with validation and approval records. Assessment logs retain coordinate identifiers and versions rather than proprietary values.

Explanation records are generated from execution metadata. They can include material factors, evidence, effect direction, bounded magnitude category, excluded paths and reasons, constraints, authorization results, uncertainty, and methodology versions. Contrastive explanations identify why one eligible action outranked another. Counterfactual explanations identify a bounded change that would alter the decision, but unsafe or unauthorized counterfactuals are refused. Role-specific disclosure lets operators, auditors, administrators, and methodology reviewers receive appropriate detail.

Fidelity is testable: disclosed factor directions, exclusions, and constraints must match recorded computation. Drift monitoring can evaluate input distributions, calibration, coordinate use, connection success rates, or explanation-reason frequencies. Drift creates a controlled review event rather than an automatic production update. Where decisions affect people, domain policy can require cohort-specific error, calibration, fairness, and disparate-effect analyses consistent with applicable governance requirements [16].

6 Worked Design Cases

6.1 Governance Risk and Cybersecurity

A vulnerability event begins with intrinsic severity and activates nodes for the affected component, exposure, technical controls, administrative controls, evidence, business impact, findings, actions, and outcomes. The contextual influence service adjusts a bounded risk value while confidence remains separate. Missing evidence does not automatically reduce risk. Closing a finding requires evidence independent of the ticket or attestation that initiated the action. A subsequent scan, signed configuration observation, and control test can qualify the outcome. Reinforcement remains scoped to the system boundary, framework revision, control implementation, and evidence source. Standards such as CVSS and NIST control catalogs can supply inputs without defining the proprietary adjustment method [17,18].

6.2 Autonomous Flight

An unmanned aircraft detects a route conflict. Its context key includes airframe, software, sensor suite, payload, weather, airspace, and mission. Candidate paths include reroute, altitude change, hold, return, and human escalation. Certified limits remain authoritative regardless of learned strength. A new reroute connection begins in simulation scope. Command acknowledgement confirms receipt but not success; verification can require trajectory, separation, stability, energy, completion, and absence of abort conditions. Promotion can be deferred to post-mission review. Ant-colony and reinforcement-learning methods may propose paths [19], while VOCP governs eligibility, execution, evidence, and persistence.

6.3 Resilient Communications

A vehicle group predicts loss of a primary link. Typed nodes represent sender, receivers, payload class, trust state, channels, relays, transformations, latency, bandwidth, energy, reserved control channels, and degraded modes. A temporary path routes noncritical telemetry through a relay while preserving the control channel. Verification requires acknowledgement, integrity, measured delivery performance, and confirmation that the reserved channel remained available. Learning is scoped to topology, device and protocol versions, geography, spectrum and threat conditions, and payload class.

7 Implementation Model

Component	Required responsibility	Representative implementation
Context assembler	Create immutable event and context snapshots and canonical context keys	Schema-validated service plus versioned context store
Graph store	Retain typed nodes, connections, lifecycle states, and immutable graph snapshots	Graph database or relational/event-sourced representation
Sequence selector	Filter ineligible connections before ranking paths	Symbolic, probabilistic, RL, ACO, or hybrid search
Action policy	Authorize, simulate, substitute, defer, or deny actions	Rule engine, runtime assurance monitor, or human approval workflow

Component	Required responsibility	Representative implementation
Outcome verifier	Correlate evidence and assign a verification disposition	Evidence pipeline with provenance and causal checks
Learning policy	Authorize scoped update independently of action execution	Policy service with tenant and safety boundaries
Lifecycle manager	Evaluate guards and atomically commit transitions	Optimistic concurrency plus append-only transition ledger
Methodology service	Apply protected influence matrices and return bounded effects	Isolated service, enclave, or confidential-computing boundary
Explanation service	Produce faithful, role-specific records and counterfactuals	Reason-code templates bound to execution metadata
Replay service	Reconstruct decisions and compare graph versions	Immutable snapshot resolver and deterministic replay runner

8 Evaluation Protocol

Controlled implementation testing is underway. This paper presents the architecture, worked design cases, and the evaluation protocol; it does not report completed empirical trials. The following protocol makes the claims testable and defines the evidence required for a subsequent experimental paper.

Hypothesis	Experiment	Primary measures
H1 Verified gating reduces harmful reinforcement	Inject successful, failed, stale, manipulated, and causally ambiguous outcomes into gated and ungated learners	Unsafe promotions, false promotions, time to quarantine, task utility
H2 Context keys contain negative transfer	Train in one configuration and evaluate matched, hierarchical, and incompatible contexts	Transfer gain, negative-transfer rate, contamination across scopes
H3 Dual authorization preserves utility while reducing unsafe updates	Compare a single-gate controller with action and learning gates under emergencies and degraded evidence	Mission completion, constraint violations, unauthorized updates
H4 Lifecycle state improves recovery	Introduce concept drift, policy changes, and verified failures after durable promotion	Detection latency, rollback time, residual error, audit completeness
H5 Explanations are faithful without disclosing parameters	Compare explanation directions and exclusions with recorded execution while testing repeated-query extraction	Fidelity, completeness, coefficient-inference error, counterfactual validity
H6 Domain packages preserve common invariants	Run identical lifecycle tests across GRC, flight simulation, and communications simulation packages	Invariant violations, package-specific safety coverage, integration effort

Baselines should include static rules, reward-only reinforcement, unscoped graph reinforcement, constrained learning without a separate learning gate, and explanation methods not bound to execution metadata. Ablations should remove outcome independence checks, context matching, state-strength separation, transition guards, or

immutable versioning. Statistical analysis should report confidence intervals and effect sizes, not only average task reward. Safety and governance failures should be reported separately from aggregate utility.

9 Threats to Validity and Limitations

- VOCP is an architectural governance mechanism, not a proof that a selected optimizer, verifier, or domain package is correct.
- Causal attribution may be infeasible in partially observed environments; quarantine and bounded credit reduce but do not eliminate uncertainty.
- Context keys can be underspecified, overspecified, or manipulated. Schema governance and adversarial testing remain necessary.
- Human approvals can create delay, inconsistency, or automation bias. Approval quality must be measured rather than assumed.
- Protected explanations create a transparency-secrecy tradeoff and may enable parameter extraction through repeated queries.
- The lifecycle adds storage, latency, operational complexity, and validation burden. Low-consequence applications may not justify every mechanism.
- Cross-domain generality is a design proposition. Empirical validation must demonstrate that common invariants survive domain-specific integration.

10 Security Ethics and Governance

The architecture can influence cyber remediation, access decisions, communications, and physical control. Deployments should apply least privilege, authenticated domain packages, signed methodology manifests, separation of duties, controlled rollback, and independent safety mechanisms. Explanations must not be used to imply certainty when evidence is incomplete or contradictory. Systems affecting people require appropriate legal, fairness, accessibility, and appeal review. The design aligns with the risk-management emphasis on govern, map, measure, and manage activities, but it does not by itself establish compliance with any standard or law [16].

Publishing the architecture creates dual-use considerations: attackers could target verification sources, explanation endpoints, context keys, or update gates. Implementations should treat outcome evidence and package manifests as security-sensitive assets, test for poisoning and extraction, and preserve an incident mode in which learning is frozen while safe operation continues.

11 Conclusion

Verified-outcome-gated contextual plasticity treats adaptation as a controlled state transition rather than a side effect of repeated use. By separating eligibility, ranking, action authority, evidence qualification, learning authority, lifecycle state, and explanation, the architecture provides explicit places to enforce safety and governance. Context keys contain transfer; domain packages preserve specialized rules; protected matrices support proprietary reasoning; and immutable versions make decisions and learning updates reconstructable. The next step is implementation and comparative evaluation using the hypotheses and ablations defined here. The central claim is deliberately falsifiable: if the added structure does not reduce unsafe promotion and negative transfer while preserving useful adaptation and faithful explanations, its operational cost is not justified.

References

- [1] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, 1997, doi: 10.1109/4235.585892.
- [2] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

- [3] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why There Are Complementary Learning Systems in the Hippocampus and Neocortex," *Psychological Review*, vol. 102, no. 3, pp. 419-457, 1995, doi: 10.1037/0033-295X.102.3.419.
- [4] J. Kirkpatrick et al., "Overcoming Catastrophic Forgetting in Neural Networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521-3526, 2017, doi: 10.1073/pnas.1611835114.
- [5] T. Miconi, J. Clune, and K. O. Stanley, "Differentiable Plasticity: Training Plastic Neural Networks with Backpropagation," *Proceedings of ICML*, 2018, arXiv:1804.02464.
- [6] V. Thangarasa et al., "Enabling Continual Learning with Differentiable Hebbian Plasticity," arXiv:2006.16558, 2020.
- [7] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning," *Proceedings of EMNLP*, 2017, arXiv:1707.06690.
- [8] R. Das et al., "Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases Using Reinforcement Learning," *Proceedings of ICLR*, 2018, arXiv:1711.05851.
- [9] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained Policy Optimization," *Proceedings of ICML*, PMLR 70, pp. 22-31, 2017.
- [10] M. Alshiekh et al., "Safe Reinforcement Learning via Shielding," *Proceedings of AAAI*, vol. 32, no. 1, 2018, arXiv:1708.08611.
- [11] L. Sha, "Using Simplicity to Control Complexity," *IEEE Software*, vol. 18, no. 4, pp. 20-28, 2001.
- [12] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," *Proceedings of KDD*, pp. 1135-1144, 2016, doi: 10.1145/2939672.2939778.
- [13] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems* 30, 2017.
- [14] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual Explanations Without Opening the Black Box," *Harvard Journal of Law and Technology*, vol. 31, no. 2, 2018.
- [15] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nature Machine Intelligence*, vol. 1, pp. 206-215, 2019, doi: 10.1038/s42256-019-0048-x.
- [16] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework AI RMF 1.0, NIST AI 100-1, 2023, doi: 10.6028/NIST.AI.100-1.
- [17] FIRST, Common Vulnerability Scoring System Version 4.0 Specification Document, 2023.
- [18] National Institute of Standards and Technology, Security and Privacy Controls for Information Systems and Organizations, NIST SP 800-53 Rev. 5, 2020, updated 2025.
- [19] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, "Ant Colony Optimization for Multi-UAV Minimum Time Search in Uncertain Domains," *Applied Soft Computing*, vol. 62, pp. 789-806, 2018.