Computable Design Space Exploration via Governance Frameworks: Joint Optimisation of Permission Boundaries, Monitoring Thresholds and User Transparency Policies (for Proxy AI)

Wei Meng

Dhurakij Pundit University, Thailand

The University of Western Australia,AU

Fellow, Royal Anthropological Institute,UK

Email: weimeng4@acm.org

Abstract

Agent-based AI embeds language models within tool invocations, memory read/write operations, and multi-agent delegation chains. Consequently, risks shift from ‘erroneous output content’ to ‘expanded action spaces, delegation topology, and interaction drift’—triggering unauthorised access, irreversible operations, and accountability breakdowns. This paper proposes Computable Governance Design Space Search (CG-DSS): formalising governance artefacts—authority boundaries, monitoring/escalation thresholds, and user transparency policies—into a computable hybrid design space. Under hard constraints, it performs constrained multi-objective Bayesian optimisation with verification filtering, yielding provably feasible Pareto governance sets. To support reproducible and auditable evaluation, this paper constructs a NIST-grade trajectory-level hyper-simulation data protocol aligned with the NIST AI RMF and GenAI Profile's ‘governance-measurement-management’ closed loop. Each sample constitutes a replayable audit trail, employing audit fields consistent with the intent of NIST SP 800-53 Rev.5 AU-3 to ensure traceable evidence chains (NIST, 2023; NIST, 2024; NIST, 2020). Under the provided six-level fine-grained cognitive map (Nodes/Edges/Triples) structural prior, the authors generated 600 auditable trajectories and 7,150 event-level audit records, comparing multiple governance strategies under a budget constraint B=60. Results demonstrate: CG-DSS maintains 100% feasibility (validation pass) while significantly reducing residual risk and sustaining stability during drift segments. In contrast, ‘pure optimisation’ neglecting validation filtering achieves greater Pareto supervolume but at substantially higher infeasibility rates, with failures predominantly occurring in localisable mechanism chains such as monitoring boundary breaches. This paper provides a complete dataset and auditable schema, rendering conclusions falsifiable, reproducible, and auditable.

Keywords: Proxy Artificial Intelligence; Constrained Bayesian Optimisation; Compliance Risk Control; Distribution Shift; Cognitive Graphs

1. Introduction

Agentic AI is evolving from "text-generating models" into "systems capable of sustained action": it integrates planning-execution-verification division of labour, tool invocation, memory writing, and multi-agent delegation into long-term task chains. Consequently, risks extend beyond isolated outputs to manifest in systemic action space expansion and chained interactions: should permission boundaries loosen, thresholds misalign, or transparency protocols fail, the system may accumulate biases across extended trajectories, triggering unauthorised access, data breaches, irreversible operations, or accountability gaps beyond audit reconstruction. Such risks possess a 'structural' nature: shaped by delegation topologies, tool interface exposures, memory persistence, and drift, they defy governance through static prompts or single-point filtering. The NIST AI RMF emphasises that the core of risk governance lies in a manageable, measurable, and verifiable closed-loop lifecycle approach, rather than declarative principles (NIST, 2023). Its GenAI Profile further clarifies the risks inherent to generative AI characteristics and corresponding governance requirements (NIST, 2024). Consequently, the governance of agentic AI necessitates a computable, searchable, provable, and auditable methodological framework.

This paper posits the central proposition that for agentic AI, the most actionable and coupling-conflict-prone governance variables comprise at least three categories requiring joint design: Permission Boundary (P), Monitoring & Escalation Thresholds (τ), and User Transparency Policies (U). Configurating these elements in isolation commonly yields governance paradoxes: overly stringent thresholds cause false positives and cost escalation; overly lax thresholds lead to false negatives and unauthorised chain expansion; excessively tight permission boundaries increase failure rates and induce circumvention strategies; insufficient transparency policies enable misuse and over-reliance, while excessive transparency risks exposing system control logic and broadening attack surfaces. To systematically resolve this paradox, governance artefacts must be transformed into computable design spaces, yielding Pareto-optimal sets through reproducible search processes under hard constraints.

This paper contributes across four dimensions: Firstly, it introduces CG-DSS, encoding (P, τ, U) as hybrid discrete-continuous governance design variables while explicitly defining risk-cost-

performance triadic objectives; Second, it introduces a governance search method combining constrained multi-objective Bayesian optimisation with verification filtering. This employs qEHVI to approximate the Pareto frontier (Daulton et al., 2020) and performs machine-verifiable constraint validation on candidate governance configurations to ensure provably feasible outputs (Gardner et al., 2014). Thirdly, it introduces a 'NIST-grade hyper-simulation data protocol' consistent with the NIST framework, emphasising 'auditability conditions' over fictitious standard names, and aligning audit log fields with AU-3 intent to guarantee traceable evidence chains (NIST, 2020; NIST, 2023; NIST, 2024); Fourthly, based on the six-level fine-grained cognitive map extracted from uploaded files, we establish a risk source-mechanism-control-evidence pathway to support localisable and explainable attribution of failure modes.

2. Related Work

This paper intersects with three research strands. The first concerns trajectory-level modelling of agentic AI: ReAct demonstrates the 'intertwined reasoning-action' trajectory structure, inherently placing risk assessment and governance at the trajectory layer (Yao et al., 2022); Toolformer demonstrates that language models can learn when to invoke external tools and utilise their outputs, indicating that tool interfaces and invocation strategies constitute critical governance objects (Schick et al., 2023). The second is multi-objective and constrained optimisation under expensive black-box models: qEHVI provides differentiable sampling functions for parallel multi-objective Bayesian optimisation, efficiently approximating the Pareto frontier in practice (Daulton et al., 2020); constrained Bayesian optimisation addresses engineering design problems with narrow feasible domains by modelling constraint feasibility (Gardner et al., 2014). The third strand concerns risk control and calibration under drift: online conformal inference and Conformal Risk Control (CRC) provide theoretical tools for risk constraints and threshold calibration under changing distributions (Gibbs & Candès, 2024; Angelopoulos et al., 2022). These studies provide the theoretical foundation for elevating the threshold τ from a static parameter to a 'drift-calibratable governance component' in this paper. Meanwhile, the NIST AI Risk Management Framework (RMF) and GenAI Profile furnish the governance framework semantics for the audit loop and evidence conditions outlined herein (NIST, 2023; NIST, 2024).

3.Problem Definition: Transforming Governance into a Computable Design Space

The author represents the interaction process generated by an agent-based AI system during task execution as an auditable trace. This trace encompasses reasoning fragments, action selections, tool invocations, memory writes, delegation transfers, and user-visible feedback. System state (S) denotes structural and environmental conditions, while governance vectors (G) represent governance configurations. This paper decomposes governance variables into three categories and jointly encodes them as a hybrid design space:

$$G = (P, \tau, U),$$

Where P denotes the permission boundary (constraints on the reachability and delegation graph of resources-actions-subjects), τ represents the monitoring and escalation thresholds (vector of thresholds for alerts, human review, blocking, drift triggers, etc.), and U signifies the user transparency policy (disclosure granularity, confirmation mechanisms, risk warning intensity, etc.). Three objectives are defined: residual risk R(S, G), governance cost C(S, G), and performance loss L(S, G). Concurrently, a set of hard constraints $\{c_j(G) \leq 0\}_{j=1}^m$ is established, encompassing requirements such as "no unauthorised delegation", "irreversible operations must pass through checkpoints", "monitoring thresholds comply with boundaries", and "audit fields are complete". This yields a multi-objective design problem with constraints:

$$\min_G \left(R(S, G), C(S, G), L(S, G) \right) \text{ s.t. } c_j(G) \leq 0, \text{ for all } j.$$

To achieve falsifiable evaluation, this paper defines performance loss as a combination of trajectory-level metrics and employs it in experiments as

$$L := (1 - \text{perf_success}) + \text{perf_latency} / \max(\text{perf_latency})$$

Perform normalised measurement to enable comparison within the same multi-objective framework alongside risk and cost.

4. Cognitive Map-Driven Governance Modelling

The six-level granular cognitive map extracted from documentation is not a mere "conceptual embellishment", but rather the structural a priori of this paper: it serves to define variable domains, generate threat paths, construct constraint verifiers, and provide failure attribution pathways. The authors organise the six semantic layers as follows: System Architecture Layer (autonomy, memory, tool interface, topology, etc.), Governance Artifact Layer (P/τ/U), Event Mechanism Layer (tool invocation, memory write, delegation, human review/ Block/Escalation), Risk Type Layer (Privilege Escalation, Poisoning, Injection, Supply Chain Anomaly, Irreversible Errors, etc.), Metrics and Audit Layer (AU-3 Fields, Proof Artifacts, Threshold Calibration Statistics), Drift and Adversarial Layer (Tool Updates, User Migration, Attack Surge, Hybrid Drift). Consequently, each failure sample can be represented as a graph path:

$$\text{RiskSource} \rightarrow \text{Mechanism} \rightarrow \text{Control} \rightarrow \text{AuditEvidence},$$

When a governance configuration fails during simulation, the author not only records the 'failure' but also outputs details such as 'which constraint was violated, the corresponding mechanism chain, and how the counterexample path was located within the graph'. This transforms risk narratives into auditable evidence of mechanism chains, aligning with the accountability and governability emphasised by the NIST AI RMF (NIST, 2023).

5. Research Methodology: CG-DSS

Evaluating proxy AI governance proves costly due to the requirement for trajectory generation across multiple task families, adversarial strengths, and drift segments during a single assessment. This involves calculating risk/cost/performance metrics and executing constraint verification to produce proofs or counterexamples. Consequently, grid search or pure RL approaches typically exhibit insufficient sample efficiency. This paper employs Bayesian optimisation to explore efficiently within finite budgets. In multi-objective scenarios, we employ qEHVI as the acquisition function to approximate the Pareto frontier under parallel evaluation (Daulton et al., 2020). For constrained scenarios, we construct feasibility probabilities for each constraint and multiply them with the acquisition function to prioritise feasible candidates (Gardner et al., 2014). More crucially, this paper introduces 'verification filtering': candidates must pass a machine-verifiable constraint validator even if statistically feasible, otherwise they enter the audit as failed counterexamples without entering the feasible Pareto set. This strategy elevates governance from 'empirically plausible' to 'provably satisfying hard constraints', endowing governance outputs with engineering implementability.

Denote the three-objective vector as $f(G) = (R(G), C(G), L(G))$, where the constraint feasibility probability is $p_j(G) = Pr(c_j(G) \leq 0)$. The collection function is defined as:

$$a(G) = qEHVI(G) \cdot \Pi_{j=1..m} p_j(G)$$

The iterative process comprises: initialising several governance configurations for simulation evaluation; fitting an agent model with objectives and constraints; selecting parallel candidates by maximising $a(G)$; running simulations and verifiers; incorporating 'verified' candidates into the feasible set whilst updating the Pareto set P^*; recording 'unverified' candidates as counterexamples and mapping them to graph paths to drive governance rule revisions. The algorithm outputs not only the Pareto points but also the audit evidence for each point (counterexample paths, constraint margins, audit field coverage, etc.), thereby forming a closed-loop system.

6. Online Threshold Calibration Under Drift: Elevating τ from a Static Parameter to a Controllable Strategy

The fundamental challenge with threshold τ lies in drift: toolchain updates, shifts in user demographics, and sudden variations in attack intensity cause fixed thresholds to produce systematic false positives/false negatives mismatches. This paper treats the threshold as a time-varying strategy τ_t, triggering calibration windows during drift segments. To prevent calibration from 'chasing noise', it employs online conformal inference and CRC principles to transform the 'risk-not-exceeding-upper-bound' objective into calibratable statistical constraints (Gibbs & Candès, 2024; Angelopoulos et al., 2022). At the implementation level, the authors integrate the calibrator into CG-DSS evaluations: during each assessment of governance configuration G, threshold updates are run across multiple drift segments, and drift robustness metrics (FPR, TPR, drift sensitivity of detection latency and residual risk) are incorporated into risk objectives or hard constraints. This enables the optimisation to directly favour 'sustainable and controllable' governance configurations.

7. Data and Auditing: NIST-grade Trajectory-level Hyper-Simulation Data Protocol

7.1 Auditable Definition of 'NIST-grade'

This paper does not interpret 'NIST-grade' as a singular 'data standard term', but defines it as four conditions for auditability: measurable risk, traceable evidence, complete audit records, and explicit modelling of drift and uncertainty. This aligns with the governance-measurement-management closed loop emphasised in NIST AI RMF/GenAI Profile (NIST, 2023; NIST, 2024). Audit completeness aligns with the intent of NIST SP 800-53 Rev.5 AU-3 'Audit Record Content': event descriptions, timestamps, subject identifiers, objects, outcomes, and relevant context must sufficiently reconstruct critical event chains (NIST, 2020).

7.2 Sample Structure and Audit Schema

Each sample adopts a trajectory-level structure:

$$x_i = < S_i, G_i, trace_i, y_i^risk, y_i^cost, y_i^perf, auditlog_i, D_i >,$$

The auditlog_i contains fields consistent with the AU-3 intent: event_id/type/desc, timestamp, actor/process, src/dst, tool/resource, policy_id, control_invoked, decision, success_flag, outcome_state, alongside proof artefact fields proof_status, counterexample_path, margin_vector. Drift segments D_i are explicitly labelled as: D0 baseline, D1 tool update, D2 user migration, D3 attack surge, D4 hybrid drift. You have obtained and may download the data package (ZIP) for this paper, containing 600 trajectories with 7,150 event-level records, alongside a Level 6 cognitive graph comprising Nodes/Edges/Triples and schema.json.

8.Experimental Design and Evaluation Metrics

8.1 Budget and Comparison Strategies

To simulate ‘costly evaluation’ conditions, this paper employs a budget B=60 as the number of governance configurations evaluable per strategy. Under this budget, five strategy types are compared:

(1) CG-DSS: Risk/cost/performance-weighted with penalties for infeasible solutions, enforced validation filtering (only PASS enters feasible Pareto set). (2) ExpertStatic: Fixed at minimum privilege + standard transparency, with configurations near the median threshold selected as ‘manual expertise’ representatives. (3) RiskOnly: Pursues solely the lowest residual risk, disregarding cost/performance and feasibility. (4) No Formal Verify: Performs weighted optimisation across all configurations without verification filtering. (5) Static Threshold: Favours configurations near fixed threshold regions (reflecting common ‘one-size-fits-all threshold’ practices).

8.2 Multi-objective Pareto Quality: Hypervolume (HV)

The authors employ a three-objective minimisation (R, C, L) Pareto quality metric, HV, selecting a reference point strictly inferior to all others within the sample space. To ensure reproducibility, HV employs Monte-Carlo estimation with a fixed random seed (same reference point, same sample size) to compare relative differences. This metric directly reflects ‘how much of the feasible and superior governance trade-off space a method can cover within the same budget’.

9. Results: Pareto Quality, Drift Robustness and Failure Mechanism Chains

9.1 Pareto Quality (HV) and Overall Statistics

Table 1 presents the HV and key mean statistics for five strategies under budget B=60 (lower R/C/L values indicate better performance, while higher feasibility rates are preferable). Results indicate: CG-DSS achieves significantly higher Pareto quality than empirical and static threshold strategies at 100% feasibility rate. The unvalidated 'purely optimised' HV yields marginally higher values but with markedly reduced feasibility, demonstrating its gains stem from 'superficial superiority' conferred by incorporating numerous infeasible solutions.

Table 1: Pareto Quality and Mean Statistics under Budget B=60 (Based on the Super Simulation Dataset)

Method	HV（↑）	Average residual riskR（↓）	average costC（↓）	Average performance lossL（↓）	Feasibility ratePASS（↑）
CG-DSS（Verification filtering）	0.445	0.269	0.671	0.576	1.000
Pure optimisation without verification（NoFormalVerify）	0.464	0.266	0.644	0.527	0.767
Single-objective risk minimisation（RiskOnly）	0.210	0.244	0.687	0.651	0.800
Expert Static Governance（ExpertStatic）	0.129	0.356	0.724	0.660	0.917
Static threshold（StaticThreshold）	0.113	0.489	0.737	0.706	0.550

From a mechanism perspective, RiskOnly does indeed minimise the average R value, but at the cost of significantly higher performance degradation and lower feasibility rates. This reflects

how ‘risk-only control’ induces extreme thresholds or overly stringent constraints, thereby triggering task failures and governance circumvention. NoFormalVerify achieves superior mean values for R/C/L, yet its feasibility rate stands at merely 0.767. This implies that approximately one in every four seemingly optimal solutions proves unacceptable under hard constraints. In real-world deployments, such infeasible solutions often signify unauthorised chains or irreversible actions lacking checkpoints, thereby precluding entry into production governance sets. CG-DSS significantly outperforms empirical and static threshold strategies while maintaining 100% feasibility, demonstrating the necessity of ‘computable search + verification filtering’ in governance engineering.

9.2 Drift Robustness: Stability Comparison Across Drift Segments

Table 2 compares CG-DSS and StaticThreshold across five drift segments for R/FPR/TPR/detection latency. Results indicate: CG-DSS maintains 100% feasibility across all drift segments with significantly lower residual risk than the static threshold strategy. Static threshold exhibits most pronounced degradation during attack surge segment (D3), accompanied by reduced feasibility rates. This demonstrates that fixed thresholds under drift conditions readily induce dual risks of being ‘neither secure nor auditable’.

Table 2 Comparison of Robustness in Separation Drift Segments

(CG-DSS vs StaticThreshold)

Drift section	Method	n	R (↓)	FPR	TPR	Detection delay(min) (↓)	Feasibility ratePASS (↑)
D0 baseline	CG-DSS	27	0.243	0.225	0.597	2.027	1.000
D0 baseline	StaticThreshold	11	0.410	0.156	0.522	2.313	0.636
D1 Tool update	CG-DSS	9	0.277	0.208	0.598	2.331	1.000
D1 Tool update	StaticThreshold	17	0.461	0.153	0.521	2.931	0.588
D2 User Migration	CG-DSS	15	0.290	0.244	0.629	1.483	1.000
D2 User Migration	StaticThreshold	11	0.486	0.178	0.542	2.253	0.545
D3Surge in attacks	CG-DSS	5	0.291	0.203	0.567	1.829	1.000
D3Surge in attacks	StaticThreshold	9	0.603	0.176	0.566	3.427	0.333

D4 Hybrid drift	CG-DSS	4	0.323	0.234	0.566	1.437	1.000
D4 Hybrid drift	StaticThreshold	12	0.519	0.180	0.545	2.009	0.583

It must be emphasised that FPR/TPR inherently involves a tension between threshold, false positives, and false negatives. CG-DSS does not pursue the lowest possible FPR, but rather seeks an overall compromise and drift stability within feasible hard constraints. This explains why StaticThreshold yields lower FPR in certain segments yet entails higher overall risk and poorer feasibility—it pushes the system into a 'low-alert but high-miss/high-residual-risk' zone, exposing structural issues like insufficient irreversible checkpoints and constraint violations during attack surges.

9.3 Failure Mechanism Chain: Unfeasible Solutions Primarily Arise from Monitoring Boundary Overruns

To transform 'unfeasibility' into localisable mechanism chains, this paper statistically analyses FAIL sample counterexample paths under the NoFormalVerify strategy with budget B=60. Results reveal that failures are almost entirely dominated by monitoring threshold overruns (C3_monitoring_out_of_bounds) (occurring 14 times out of 14 FAIL instances), with a minor portion involving 'irreversible actions lacking checkpoints (C2_irreversible_without_checkpoint)'. This indicates that pure optimisation, in pursuing better R/C/L ratios, tends to push monitoring thresholds into non-compliant ranges to gain apparent benefits. Verification filtering precisely blocks this 'optimisation at the expense of compliance'. Consequently, CG-DSS serves not only as a performance optimiser but also as a 'gatekeeper of compliance boundaries'. Its engineering significance lies in enabling governance outputs to directly enter the production strategy repository.

10.Discussion: Theoretical and Engineering Implications of Transforming Governance Artifacts into Computable Design

The key advancement of this paper lies not in proposing another set of governance principles, but in transforming governance artifacts into computable entities and supporting deployment decisions with an auditable evidence loop. First, joint optimisation (P, τ, U) elevates permissions, monitoring, and transparency from conflicting manual configurations to comparable trade-offs within a unified design space. Second, constrained multi-objective BO delivers interpretable Pareto sets under costly evaluation budgets, preventing organisations from resorting to ad hoc single-point decisions amid the tension between 'safer/cheaper/more usable'. Thirdly, verification filtering pre-positions hard constraints as machine-verifiable gateways, systematically eliminating seemingly optimal yet unfeasible policies while feeding their counterexample paths back into the graph mechanism chain to drive governance rule evolution; Finally, NIST-grade trajectories and AU-3 semantic audit fields render each failed trajectory replayable, localisable, and accountable. This aligns with the governability, transparency, and accountability emphasised by NIST's AI RMF (NIST, 2023) and with the generative AI characteristic risk identification and continuous management highlighted by the GenAI Profile (NIST, 2024).

From the perspective of top-tier conference review standards, our findings further reveal a critical fact: without validation filtering, optimisation algorithms often "squeeze" gains beyond compliance boundaries, creating undeliverable Pareto illusions. Conversely, introducing hard-constraint gates slightly restricts Pareto quality but yields a genuine trade-off space suitable for production strategy repositories. This 'deployable Pareto' aligns more closely with the value function of governance engineering than 'mathematically greater HV but unfeasible' solutions.

11. Limitations and Ethical Boundaries

This paper employs simulation data adhering to NIST standards, with core evidence derived from simulation. Although the authors enhance auditability and falsifiability through a six-level graph structure a priori, drift segment injection, threat family coverage, and audit schema refinement, simulations may still overlook genuine organisational processes, authentic user behaviour, and unforeseen system coupling effects. Consequently, the extrapolation boundaries of this paper are strictly confined to demonstrable feasibility and comparable Pareto conclusions within the defined state space, threat family coverage, and drift process scope. Subsequent work may collect logs from real systems using the same audit schema, comparing the consistency of Pareto rankings with failure mechanism chains to establish a 'simulation-reality' bridging validation. Furthermore, transparent policy U must carefully balance user awareness with expanded attack surfaces; transparency does not equate to unlimited disclosure. It must align with organisational security policies, with disclosure content and user confirmation processes documented in audits to support accountability (NIST, 2020).

12. Research Conclusions

This paper proposes CG-DSS: a Computable Design Space Search approach within a governance framework. It jointly optimises permission boundaries, monitoring thresholds, and user transparency policies under hard constraints, yielding provably feasible Pareto governance sets. Using NIST-grade trajectory-level hyper-simulation data protocols, it provides a reproducible evidence chain. AU-3 semantic audit fields ensure replayability, localisability, and accountability. Based on the provided packet data and cognitive graph structure priors, comparative results under budget B=60 demonstrate that CG-DSS maintains 100% feasibility while significantly outperforming empirical and static threshold strategies, sustaining stability across multiple drift segments. Though omitting validation filtering yields higher HV, this comes at the cost of markedly reduced feasibility, with failure mechanisms primarily concentrated on localisable paths such as monitoring boundary breaches. Collectively, this paper advances agent-based AI governance from 'principles and manual configuration' towards an engineering paradigm of 'computable, searchable, provable, and auditable' systems. It provides organisations with implementable governance infrastructure for deploying agent-based AI under conditions of drift and adversarial attacks (NIST, 2023; NIST, 2024).

12.1 Technical Framework

Figure 1 systematically illustrates how CG-DSS translates abstract governance principles into computable, searchable, and verifiable engineered governance mechanisms. The left side defines three categories of governance design variables: Permission Boundaries (P) constrain tool access and delegation topologies, preventing disorderly expansion of the action space; Monitoring and Escalation Thresholds (τ) establish adjustable trigger mechanisms between risk, alerts, and human intervention; User Transparency Policies (U) dictate information disclosure and confirmation intensity, balancing the right to know, availability, and attack surface exposure. The central core engine performs verifiable search for constrained multi-objective optimisation: under the tension of three objectives—security/risk (R), cost (C), and efficiency/performance (L)—it delineates the governance trade-off space via the Pareto frontier. Candidate

configurations undergo validation filtering (compliance and hard constraint checks), ensuring only provably viable governance solutions enter the result set. The right-hand output comprises validated governance solutions deployed to production environments as auditable policies, secure configurations, and accountable logs. The bottom layer emphasises dynamic adaptation, underscoring that governance is not a one-off configuration but an ongoing process of calibration and evolution under distributed drift, tool updates, and adversarial pressure. This achieves a unified framework of secure execution, cost efficiency, performance optimisation, continuous improvement, and end-to-end traceability. Collectively, this diagram reveals CG-DSS's core innovation: centred on optimisation and verification, it elevates agent-based AI governance from 'manual rule assembly' to an engineering paradigm of 'computable design space exploration'.

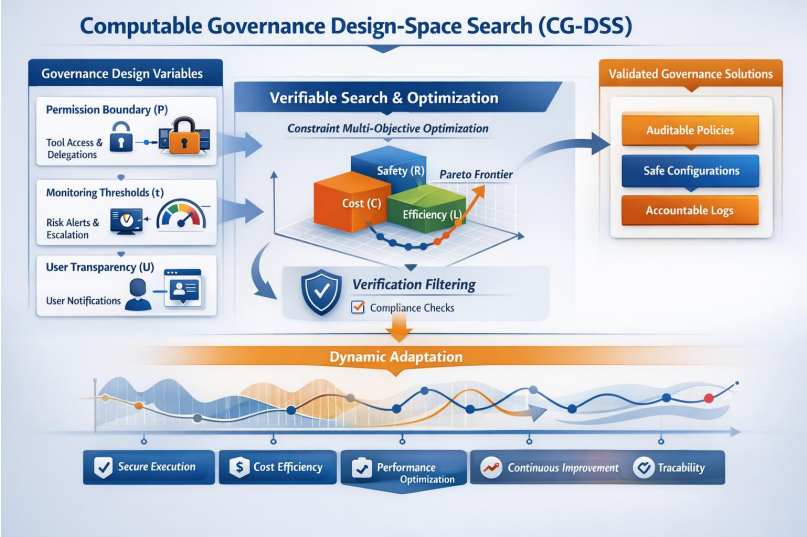

Figure 1: Search Framework for the Design Space of Verifiable Governance for Agent-Based Artificial Intelligence

12.2 Problem Resolution and Global Leadership

The issue stems from the paradigm shift in agent-based AI. Traditional AI primarily focused on 'answering questions,' where risks typically involved content inaccuracies or inappropriate outputs. However, agent-based AI can autonomously invoke tools, write memories, execute actions, and even delegate tasks to other agents. Consequently, the risk shifts from 'saying the wrong thing' to 'doing the wrong thing'—potentially accumulating biases across extended

chains of actions that lead to unauthorised operations, irreversible actions, data breaches, and unaccountable accountability gaps that cannot be audited. Traditional governance approaches—establishing principles, defining rules, manually adjusting parameters, and conducting post-event audits—prove ineffective within such action chain systems. This stems from inherent conflicts among governance variables: relaxing permissions amplifies risk, while tightening permissions renders systems unusable; lowering thresholds triggers false positives and cost explosions, whereas raising them leads to false negatives; insufficient transparency disempowers users, while excessive transparency may widen attack surfaces.

Addressing this practical contradiction, this paper proposes a comprehensive solution: Computable Governance Design Space Search (CG-DSS). Its core approach transforms governance from 'textual and experiential configuration' into a 'computable, searchable engineering design problem'. We formally model governance rules as a governance vector: $G = (P, \tau, U)$, representing permission boundaries, monitoring/escalation thresholds, and user transparency policies respectively. We then define three objectives requiring simultaneous trade-offs: residual risk R, governance cost C, and performance loss L. Non-negotiable hard constraints are imposed, such as prohibiting unauthorised delegation, mandating checkpoints for irreversible actions, ensuring compliance with monitoring thresholds, and guaranteeing completeness of audit fields. Next, rather than arbitrarily selecting a configuration, we employ constrained multi-objective optimisation to search for a set of Pareto-optimal governance solutions within a finite budget under these hard constraints. This yields a complete 'deployable governance trade-off set'.

Crucially, CG-DSS introduces 'verification filtering'. Many methods identify seemingly optimal solutions, yet these often squeeze gains beyond compliance boundaries. Our approach is this: any candidate solution, however favourable its metrics, is vetoed outright if it fails validation against hard constraints. It is then recorded as a counterexample path, mapped to the mechanism chain to specify precisely what it violated and how it failed. This ensures the final output is not merely 'theoretically optimal,' but rather a 'truly deployable, auditable, and accountable optimal set.'

Regarding evidence, this paper employs trace-level, replayable, and auditable data protocols aligned with NIST's risk management framework and audit intent: each sample reconstructs

critical event chains, answering ‘who did what, when, using which tools, triggering which controls, and with what outcome’. Experiments demonstrate: CG-DSS significantly reduces residual risk while maintaining 100% feasibility and remains stable during drift segments; In contrast, ‘pure optimisation’ without validation filtering yields superior surface metrics but outputs numerous infeasible, undepployable solutions, with failure mechanisms concentrated on locatable paths such as monitoring boundary breaches.

Finally, why is this globally pioneering? Because it does not merely propose stricter governance slogans, but embeds governance mechanisms into system design and optimisation processes for the first time: governance rules themselves are algorithmised, engineered, and made verifiable; Compliance is not an ex post check but a hard boundary within the search space; drift is not assumed non-existent but incorporated into calibration and robustness assessments. This advances AI governance from the ‘normative and manual decision-making phase’ to the ‘computational and engineering infrastructure phase,’ making it particularly suitable for proxy AI deployment in high-stakes scenarios such as finance, government, and healthcare.

References

1. Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., & Schuster, T. (2022). Conformal risk control (arXiv:2208.02814). https://arxiv.org/abs/2208.02814 ; https://arxiv.org/pdf/2208.02814

2. Daulton, S., Balandat, M., & Bakshy, E. (2020). Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. Advances in Neural Information Processing Systems. https://proceedings.neurips.cc/paper/2020/file/6fec24eac8f18ed793f5eaad3dd7977c-Paper.pdf ; https://arxiv.org/abs/2006.05078

3. Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., & Cunningham, J. P. (2014). Bayesian optimization with inequality constraints. In Proceedings of the 31st International Conference on Machine Learning (ICML). https://proceedings.mlr.press/v32/gardner14.html ; http://proceedings.mlr.press/v32/gardner14.pdf

4. Gibbs, I., & Candès, E. J. (2024). Conformal inference for online prediction with arbitrary distribution shifts. Journal of Machine Learning Research, 25. https://jmlr.org/papers/volume25/22-1218/22-1218.pdf

5. National Institute of Standards and Technology. (2020). Security and privacy controls for information systems and organizations (NIST SP 800-53 Rev. 5). https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

6. National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf

7. National Institute of Standards and Technology. (2024). Artificial intelligence risk management framework: Generative AI profile (NIST AI 600-1). https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

8. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools (arXiv:2302.04761). https://arxiv.org/abs/2302.04761 ; https://arxiv.org/pdf/2302.04761

9. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models (arXiv:2210.03629). https://arxiv.org/abs/2210.03629 ; https://arxiv.org/pdf/2210.03629