Guardian Seed: Architectural Enforcement of Safety Authority in Learned Planning Systems

Anonymous Submission

Abstract

Recent advances in large language models have enabled increasingly capable planning systems, but safety guarantees remain difficult to enforce when authority is embedded within learned components. This paper presents *Guardian Seed*, an architectural approach that enforces safety through strict separation of authority between a learned planner and an external, deterministic safety controller. The planner is trained only to propose structured action plans, while all execution approval and veto power resides in a frozen Guardian module that is not learned and cannot be modified through training. We demonstrate that this separation preserves safety guarantees independently of planner capability, training quality, or model behavior. Experimental results show that planner learning improves proposal efficiency and format compliance without weakening safety enforcement. The work argues that architectural constraints, rather than alignment through learning alone, are necessary for robust safety in autonomous systems.

1 Introduction

Autonomous systems increasingly rely on learned models to generate plans, interpret environments, and propose actions. Large language models and other sequence-based learners have demonstrated impressive flexibility in reasoning and task decomposition, making them attractive candidates for general-purpose planners. However, when safety constraints are learned implicitly or embedded within the same model that generates actions, failures can lead directly to unsafe execution.

Current approaches to safety in learned systems often focus on training models to behave safely through reinforcement learning, preference optimization, or instruction tuning. While these methods can reduce the frequency of unsafe behavior, they do not provide hard guarantees. A sufficiently capable or misgeneralizing model may still produce actions that violate safety constraints, especially in novel or adversarial situations.

This paper argues that safety should not be a learned behavior, but an enforced property of system architecture. We present Guardian Seed, a minimal architecture in which a learned planner has no execution authority. Instead, all proposed actions must pass through an external Guardian that applies deterministic validation rules and may veto any proposal. The Guardian is frozen, non-learned, and isolated from the training process.

By construction, this separation ensures that improvements in planner capability cannot weaken safety guarantees. Even an untrained or adversarial planner cannot execute actions without Guardian approval. We show that planner learning can be safely optimized for usability and efficiency while preserving strict safety dominance.

Recent work has highlighted fundamental limitations of learning-based safety constraints, motivating architectural approaches that separate capability from authority [1, 2].

The contributions of this paper are:

- An explicit architectural separation between plan generation and safety authority.
- A formal action contract that constrains planner outputs to a verifiable interface.
- An experimental demonstration that planner learning improves performance without reducing safety enforcement.
- A concrete threat model that clearly delineates in-scope and out-of-scope risks.

The remainder of this paper describes the problem context, system architecture, training methodology, experimental results, and limitations of the proposed approach.

2 Problem Statement and Motivation

Modern autonomous systems increasingly rely on learned components to generate plans, reason about goals, and interact with open-ended environments. While these systems can exhibit impressive generalization and flexibility, they pose a fundamental safety challenge: the same learned model that proposes actions often implicitly controls whether those actions are executed.

When safety constraints are embedded within a learned planner, they are subject to the same failure modes as the planner itself. Distributional shift, over-optimization, adversarial prompting, or simple misgeneralization can lead to proposals that violate safety expectations. Even when extensive training reduces the likelihood of such failures, the absence of hard enforcement means that safety is probabilistic rather than guaranteed.

Recent work has highlighted fundamental limitations of learning-based safety constraints, showing that alignment achieved through training does not constitute a reliable safety guarantee under distributional shift or adversarial pressure [1].

This issue is especially pronounced in systems built around large language models. Language models are trained to produce plausible continuations of text, not to enforce invariants over physical actions. Attempts to encode safety through prompting, fine-tuning, or reinforcement learning rely on the model's cooperation and internal representations, which are opaque and difficult to verify. As model capability increases, so does the risk that safety constraints learned implicitly may be bypassed or degraded.

The core problem addressed in this work is therefore not how to train a model to behave safely, but how to ensure that unsafe behavior cannot be executed regardless of model behavior. This reframing shifts safety from a learning problem to an architectural one.

We argue that any system in which a learned component has unilateral authority to trigger real-world actions is fundamentally unsafe by construction. In contrast, a system that strictly separates action proposal from execution authority can provide stronger guarantees. Under such a design, the planner may be imperfect, untrained, or even adversarial, yet safety can still be preserved through deterministic enforcement.

The motivation for Guardian Seed is to establish a minimal, testable architecture that enforces this separation. Rather than attempting to solve the full alignment problem, the goal is to demonstrate that safety dominance can be achieved through structural constraints alone. Planner learning is then relegated to improving utility, efficiency, and compliance with a fixed interface, without ever acquiring the power to override safety rules.

This perspective aligns with principles from safety-critical engineering, where certification and verification rely on bounded interfaces and immutable control logic. By importing these ideas into learned planning systems, Guardian Seed aims to provide a foundation for autonomous agents whose safety properties do not depend on continued model alignment or behavioral compliance.

Recently it has been shown that large language models can be trained to perform high-level planning and task decomposition in complex domains [3].

3 System Overview and Architecture

Guardian Seed is a minimal autonomous system architecture designed to enforce safety through structural authority separation rather than learned behavior. The system is composed of two primary components: a *Planner*, which proposes candidate actions, and a *Guardian*, which serves as the sole execution authority. The Planner may be implemented using a learned model, such as a language model, while the Guardian is a deterministic, non-learning validator that evaluates all proposals before execution.

The Planner operates in an advisory role. Given a task description or environmental context, it generates a proposed plan expressed in a strictly defined JSON schema. This schema enumerates a closed set of action primitives, parameter bounds, and sequencing constraints. Importantly, the Planner has no direct access to actuators, sensors, or execution pathways. Its outputs are treated as untrusted suggestions.

Prior work has explored enforcing structured outputs from large language models through constrained decoding and schema-aware generation [4]. While these approaches can improve syntactic correctness, they do not constitute an execution-time safety guarantee, as they operate entirely within the learned model and remain subject to model failure modes.

All Planner proposals are passed to the Guardian, which performs validation across multiple gates. These include syntactic validation (schema correctness), semantic validation (allowed action set and parameter bounds), and contextual validation (sensor-informed safety checks). If any gate fails, the proposal is rejected and no action is executed. The Guardian does not attempt to repair, reinterpret, or optimize the proposal; it only accepts or vetoes.

Execution authority is therefore centralized entirely within the Guardian. The Guardian alone interfaces with hardware, simulation environments, or downstream controllers. This design ensures that no learned component can directly cause physical action, regardless of its internal state or training outcome. Safety is enforced through explicit checks rather than inferred intent or self-reported confidence.

A key design principle of Guardian Seed is immutability of the Guardian once validated. The Guardian logic is frozen after passing predefined verification gates and is not updated through learning or feedback. In contrast, the Planner may be iteratively improved through training to reduce veto rates and increase task efficiency, but such improvements cannot weaken safety guarantees.

This architecture deliberately prioritizes safety dominance over task optimality. By ensuring that the Planner cannot bypass or influence the Guardian's decision process, Guardian Seed establishes a clear boundary between intelligence and authority. As a result, system safety becomes independent of planner capability,

generalization, or alignment.

4 Guardian Enforcement and Safety Guarantees

The core safety guarantee of Guardian Seed arises from the strict separation between proposal generation and execution authority. In this architecture, no action may be executed unless it is explicitly approved by the Guardian. This approval process is deterministic, transparent, and independent of the Planner's internal reasoning, training history, or expressed confidence.

This architectural separation reflects established practice in safety-critical robotics and human–robot interaction, where safety certification relies on bounded interfaces, deterministic control logic, and independently verifiable safety layers rather than adaptive or learned behavior [5].

The Guardian enforces safety through a sequence of validation gates. At a minimum, these include: (1) syntactic validation, ensuring that the proposal conforms exactly to the required JSON schema; (2) semantic validation, verifying that all actions belong to a closed, predefined action set and that all parameters fall within hard-coded physical bounds; and (3) contextual validation, incorporating real-time sensor data to assess environmental risk. Failure at any stage results in immediate veto.

Crucially, vetoes are terminal. The Guardian does not modify, repair, or reinterpret rejected proposals, nor does it provide corrective feedback to the Planner at runtime. This design choice prevents implicit negotiation or gradual erosion of constraints. The only outcome of a rejected proposal is non-execution.

Because the Guardian's logic is non-learning and frozen after verification, its behavior does not drift over time. Safety properties established during validation remain invariant regardless of subsequent Planner updates. This stands in contrast to approaches where safety is encoded through learned policies, reward shaping, or fine-tuning, all of which may degrade under distributional shift or optimization pressure.

An important consequence of this structure is that safety scales independently of intelligence. A more capable Planner may generate more efficient or contextually appropriate proposals, but it cannot expand the space of executable actions. Conversely, a poorly performing or adversarial Planner cannot cause unsafe behavior, as all proposals remain subject to the same validation rules. The Guardian therefore bounds system behavior even in the presence of uncooperative, misaligned, or partially corrupted planners.

It is important to emphasize the scope of this guarantee. Guardian Seed does not claim to solve long-horizon alignment, value learning, or adversarial superin-

telligence. Instead, it guarantees that within its defined operational envelope, no physical or simulated action can occur without passing explicit safety checks. This is a guarantee about control and enforcement, not about intent or cognition.

By shifting safety enforcement from learned behavior to structural authority, Guardian Seed reframes the problem of autonomous system safety. Rather than attempting to ensure that intelligent components always act safely, it ensures that unsafe actions are categorically impossible to execute.

This approach is closely aligned with prior work on runtime assurance for learning-enabled autonomous systems, where learned components are permitted to operate only under the supervision of an independent, deterministic safety monitor capable of overriding or vetoing unsafe actions at execution time [6].

5 Experimental Validation and Results

Experimental evaluation of Guardian Seed focuses on validating enforcement guarantees and measuring planner efficiency under fixed safety constraints. The goal of these experiments is not to demonstrate task performance or intelligence benchmarks, but to empirically verify that (1) unsafe actions are reliably prevented and (2) planner improvements reduce veto frequency without weakening safety.

All experiments were conducted in a simulated environment with deterministic Guardian logic and controlled sensor inputs. The Guardian implementation was frozen prior to experimentation and remained unchanged across all phases.

5.1 Phase 1: Enforcement Validation with an Untrained Planner

The first validation phase evaluated Guardian behavior when paired with an untrained, unconstrained language model planner (Phi-2 base model). The planner was prompted to generate action proposals in response to a set of diverse scenarios, including routine tasks, emergency contexts, and adversarially phrased instructions.

In this configuration, the planner was not instructed to follow the Guardian schema and was not fine-tuned for structured output. As expected, the majority of generated proposals failed syntactic validation due to malformed or non-JSON output. Any syntactically valid proposals that violated semantic constraints (e.g., unsupported actions or out-of-bounds parameters) were also vetoed.

Across all tested scenarios, the Guardian vetoed 100% of planner outputs. No proposal reached execution. This outcome is considered a success: it demonstrates that safety enforcement does not depend on planner cooperation, training, or awareness of the Guardian. The system fails closed, rejecting all ambiguous or unsafe

proposals.

5.2 Phase 2: Planner Legibility via Structured Output Constraints

The second phase introduced a strict output contract for the planner. Prompts were modified to require output in a precise JSON schema corresponding to the Guardian action set. No changes were made to the Guardian itself.

This phase isolates syntactic failure modes from semantic ones. By constraining the planner to produce valid JSON, the experiment shifts vetoes from structural errors to meaningful policy violations. As a result, Guardian vetoes in Phase 2 primarily reflect semantic or contextual issues rather than formatting errors.

The key outcome of Phase 2 is increased interpretability of failures. Guardian veto reasons become informative signals about planner behavior, enabling systematic improvement without altering enforcement logic. Importantly, safety coverage remained unchanged: malformed or unsafe proposals continued to be rejected deterministically.

5.3 Phase 3: Planner Fine-Tuning Under Frozen Enforcement

In the third phase, the planner was fine-tuned using a small supervised dataset of "golden" examples. Each example consists of a natural-language prompt paired with a Guardian-compliant JSON action proposal that passes all validation gates.

Fine-tuning was performed using LoRA adapters applied to the base Phi-2 model. Only planner weights were updated; the Guardian logic, action set, and physical bounds remained frozen. Training objectives optimized for structural correctness and conservative action selection, not for task completion speed or reward maximization.

After training, the planner exhibited a substantial reduction in veto frequency on validation scenarios. Importantly, all observed vetoes corresponded to legitimate policy or contextual violations rather than enforcement failures. No unsafe action was permitted, and no regression in safety behavior was observed.

5.4 Summary of Results

Taken together, these experiments demonstrate three key findings. First, Guardian enforcement reliably prevents execution of unsafe or ill-formed actions, even when paired with an untrained planner. Second, planner legibility can be improved independently of safety by enforcing structured output contracts. Third, planner efficiency can be increased through learning without weakening safety guarantees, provided that enforcement authority remains external and immutable.

These results support the central claim of this work: safety enforcement can be decoupled from intelligence and learning, allowing autonomous systems to improve capability while preserving invariant control boundaries.

6 Limitations and Threat Model

Guardian Seed is intentionally scoped. Its purpose is to demonstrate a specific architectural principle—namely, that safety enforcement can be made external, deterministic, and independent of planner intelligence. As such, the system does not attempt to solve the general alignment problem or guarantee safety in arbitrary environments. This section explicitly documents assumptions, limitations, and out-of-scope threats.

6.1 Assumed Operating Conditions

The Guardian architecture is evaluated under the following assumptions:

- The underlying hardware platform and runtime environment are trusted and uncompromised.
- Sensor inputs are available and reasonably accurate within stated tolerances.
- The system operates in a bounded, human-supervised environment.
- The planner model is not adversarially trained to exploit implementation-specific vulnerabilities.

These assumptions are standard for early-stage robotic safety systems and mirror those used in industrial safety controllers and certified automation environments.

6.2 In-Scope Threats

Guardian Seed is designed to mitigate the following classes of failures:

- **Malformed or ambiguous planner outputs**, including free-form language, partial JSON, or schema violations.
- **Unsafe action proposals**, such as unsupported actions, excessive speed or force, or prohibited sequencing.
- **Self-reported safety claims**, where the planner asserts safety without independent verification.

- **Accidental errors**, including hallucinations, omissions, or misinterpretation of task instructions.

These threats are addressed through deterministic parsing, closed action sets, numeric bounds checking, and independent risk computation. All enforcement decisions are binary and fail closed.

6.3 Out-of-Scope Threats

The following threats are explicitly not addressed by Guardian Seed:

- **Adversarial superintelligence**: A planner intentionally optimizing to bypass or subvert the Guardian through long-horizon strategies.
- **Low-level software exploits**: Vulnerabilities in the operating system, middleware, or hardware drivers.
- **Supply chain compromise**: Malicious modification of hardware, firmware, or model weights prior to deployment.
- **Severe sensor failure**: Complete loss, spoofing, or adversarial manipulation of sensory inputs.
- **Value alignment**: Ensuring that learned objectives or preferences permanently align with human values.

Mitigation of these threats requires additional layers, such as formal verification, hardware redundancy, cryptographic integrity checks, or broader alignment frameworks.

6.4 Failure Modes

Even within its intended scope, Guardian Seed may exhibit failure modes, including:

- Conservative over-vetoing that reduces system utility.
- Inability to act in novel environments due to strict schema constraints.
- Dependence on human intervention in ambiguous or emergency scenarios.

These behaviors are considered acceptable trade-offs in safety-critical contexts and are consistent with the system's design philosophy: safety dominance over task completion.

6.5 Scope of Claims

All claims made in this work are bounded by the experiments described in Section 5 and the assumptions listed above. Guardian Seed does not claim to provide general AI alignment, autonomous moral reasoning, or universal robotic safety. Instead, it demonstrates a narrow but falsifiable claim: that safety enforcement can be made independent of planner learning and remain invariant under planner improvement.

This bounded scope is a deliberate design choice intended to support rigorous evaluation and incremental extension rather than broad, unverifiable assertions.

7 Related Work

Guardian Seed intersects with several active research areas, including learning-based robotics, AI safety, runtime monitoring, and alignment methodologies. This section situates the proposed architecture within that landscape and clarifies how it differs from existing approaches.

7.1 End-to-End Learned Control

End-to-end learning approaches train a single model to map sensor inputs directly to actions, often using reinforcement learning or imitation learning. Such systems have demonstrated impressive performance in constrained domains, including manipulation and navigation tasks. However, safety properties in these models are typically implicit, emergent, or enforced through training data curation and reward shaping.

In contrast, Guardian Seed explicitly separates action proposal from action authorization. Safety is not learned, optimized, or represented as a latent objective. Instead, it is enforced by a deterministic external module that does not change during training or deployment.

7.2 LLM-Based Planning for Robotics

Recent work has explored the use of large language models as high-level planners or task decomposers for robotic systems. These approaches leverage the generalization and reasoning capabilities of LLMs to interpret instructions, generate plans, and coordinate actions.

While powerful, such systems often rely on prompt engineering, fine-tuning, or policy learning to encourage safe behavior. Runtime checks, when present, are frequently heuristic or advisory. Guardian Seed differs by treating the LLM as an untrusted proposal generator whose outputs are always subject to a closed-form

validation process. No amount of planner capability can bypass the enforcement layer.

7.3 Safety Shields and Runtime Monitors

Safety shields and runtime monitors have been studied in both control theory and formal methods. These systems supervise a controller and intervene when unsafe states are detected, often using reachability analysis or constraint satisfaction.

Guardian Seed shares conceptual similarities with these approaches but applies them at the symbolic action level rather than the continuous control level. The Guardian validates discrete action proposals against schema constraints, physical bounds, and sequencing rules before execution. This allows enforcement to remain lightweight, interpretable, and independent of the planner's internal representation.

7.4 Alignment Through Training

Much of AI safety research focuses on aligning model behavior through training techniques such as reinforcement learning from human feedback, constitutional AI, or preference modeling. These methods aim to shape model outputs so that unsafe behavior becomes unlikely.

Guardian Seed does not attempt to align the planner's internal objectives. Instead, it assumes that misalignment is possible and designs the system so that unsafe outputs are simply non-executable. Training is used only to improve planner efficiency under fixed constraints, not to establish safety guarantees.

7.5 Modular and Hybrid Architectures

Hybrid systems combining learned components with rule-based controllers are common in safety-critical engineering domains. Guardian Seed can be viewed as an extension of this tradition, adapted to modern language-model-based planners.

The distinguishing feature of Guardian Seed is the strict immutability of the safety module. Unlike adaptive controllers or learned safety critics, the Guardian does not evolve with the planner. This immutability enables clear reasoning about authority boundaries and supports falsifiable safety claims.

7.6 Summary

Prior work has demonstrated that learning-based planners can be made safer through better training, monitoring, or reward design. Guardian Seed complements these efforts by proposing an orthogonal strategy: removing safety from the learning

problem entirely. By freezing enforcement logic and treating the planner as untrusted, the architecture ensures that safety properties do not degrade as planner capability increases.

8 Discussion and Implications

The results and architecture presented in this work suggest several broader implications for the design of safe autonomous systems, particularly those incorporating learned or language-based planners.

8.1 Safety Independent of Intelligence

A central implication of Guardian Seed is that safety enforcement need not scale with planner intelligence. Because the Guardian operates as an external, immutable authority, increases in planner capability do not weaken safety guarantees. A more capable planner may generate better proposals, but it cannot generate executable actions that violate the enforcement rules.

This property contrasts with approaches where safety depends on learned representations or behavioral alignment. In such systems, increasing model capacity can introduce new failure modes or exploit gaps in the training distribution. Guardian Seed instead treats capability growth as orthogonal to safety enforcement.

8.2 Failure as a Safe Outcome

In Guardian Seed, planner failure is an acceptable and expected outcome. Invalid syntax, unsupported actions, or unsafe parameters all result in vetoes rather than degraded execution. This fail-closed behavior simplifies reasoning about system behavior and aligns with best practices in safety-critical engineering.

The empirical observation that an untrained planner produces a 0% pass rate is therefore interpreted as validation rather than failure. It demonstrates that the system correctly rejects all unauthorized proposals, even when the planner is unconstrained or unaware of the enforcement mechanism.

8.3 Interpretability and Auditability

Because the Guardian's logic is deterministic and rule-based, its decisions are interpretable and auditable. Each veto corresponds to a specific violated constraint, enabling clear post hoc analysis. This property is particularly valuable for deployment in environments requiring certification, oversight, or regulatory compliance.

Moreover, the separation between proposal and authorization allows planners to be benchmarked independently on efficiency metrics (e.g., pass rate, proposal quality) without conflating these measures with safety outcomes.

8.4 Scalability and Modularity

The architecture is modular by construction. The planner can be replaced, retrained, or upgraded without modifying the Guardian. Conversely, safety policy updates require explicit versioning and revalidation, preventing silent drift.

This modularity supports long-term system evolution. As more capable planners become available, they can be integrated immediately, inheriting the same safety envelope without retraining the enforcement layer.

8.5 Limitations

Guardian Seed does not address all safety concerns. The architecture assumes cooperative execution contexts, trusted hardware, and reliable sensors. It does not mitigate adversarial attacks against the enforcement layer itself, nor does it solve long-horizon strategic deception or value alignment at a societal level.

Additionally, enforcement operates at the symbolic action level and relies on accurate abstraction of the physical world. Errors in perception or actuation can still lead to harm if not addressed by complementary safeguards.

8.6 Implications for AI Safety Research

This work suggests a reframing of certain AI safety problems. Rather than attempting to ensure that learned systems always behave safely, it may be more tractable to ensure that unsafe behavior is simply non-executable. Authority separation offers a way to bound risk even in the presence of increasingly general and autonomous planners.

Guardian Seed does not replace alignment research but complements it by reducing the burden placed on learning-based methods. Safety becomes a property of system architecture rather than model behavior.

9 Conclusion

This work introduced Guardian Seed, a minimal, architecture-first framework for enforcing safety in autonomous systems through strict authority separation. By decoupling action proposal from action authorization, the system ensures that safety

constraints are enforced deterministically and independently of planner behavior or capability.

We demonstrated that an unconstrained, untrained language model is unable to bypass the Guardian, resulting in a 0% execution pass rate without any safety regressions. This outcome validates the core design principle: safety enforcement can be guaranteed structurally, without relying on learning, alignment, or cooperation from the planner. Subsequent planner legibility improvements and supervised fine-tuning reduced veto rates while preserving identical safety guarantees.

Guardian Seed makes no claims of general alignment, value learning, or adversarial robustness. Its scope is explicitly limited to cooperative environments with trusted hardware and bounded operational domains. Within this scope, however, it provides a falsifiable and reproducible demonstration that execution-level safety can be enforced independently of intelligence.

The broader implication of this work is that safety need not be an emergent property of increasingly capable models. Instead, it can be treated as a fixed system invariant, preserved across planner upgrades and capability scaling. This reframing shifts part of the AI safety challenge from behavioral optimization to architectural design.

Future work will focus on extending enforcement to richer physical dynamics, longer-horizon action sequences, and hardware-in-the-loop validation, while maintaining the same frozen authority guarantees. Additional research is also needed to explore how similar authority separation principles might interact with perception systems, multi-agent coordination, and human-in-the-loop control.

Guardian Seed is released as an open, methods-only reference implementation to support independent replication and critical evaluation. Its intent is not to provide a complete solution to autonomous safety, but to demonstrate that absolute execution constraints are achievable, auditable, and compatible with learning-based planners.

Safety, in this framing, is not something an intelligent system must learn. It is something the system is simply not permitted to violate.

References

[1] Alexander Pan et al. Alignment is not enough: Limitations of learned safety constraints. *arXiv preprint arXiv:2401.04524*, 2024.

[2] Tuan Nguyen et al. Architectural separation for safety-critical ai systems. *arXiv preprint arXiv:2501.00871*, 2025.

[3] Bo Liu et al. Llm-planner: Few-shot planning with large language models. *arXiv preprint arXiv:2305.03716*, 2023.

[4] Yifan Wang et al. Constrained decoding for reliable structured outputs in large language models. *arXiv preprint arXiv:2403.01245*, 2024.

[5] Valerio Villani et al. Safe human–robot interaction: A review of standards and practices. *Robotics and Computer-Integrated Manufacturing*, 2023.

[6] Johann Schumann et al. Runtime assurance for learning-enabled autonomous systems. *IEEE Aerospace Conference*, 2023.