Artificial Constitutionalism: A Self-Synthesizing Prompt Governance Compiler (ACGS-PGP) Framework for Advanced AI Systems

Alexei Vanguard
Institute for Advanced AI Governance
Metropolis, AI, Utopia
a.vanguard@iaag.edu

Eleanor Praxis
Centre for Verifiable Autonomy
Technopolis, AI, Utopia
e.praxis@cva.tech

Corin Synthetica
Institute for Advanced AI Governance
Metropolis, AI, Utopia
c.synthetica@iaag.edu

ABSTRACT

The rapid advancement of Artificial Intelligence (AI) necessitates robust, adaptive, and verifiable governance mechanisms. Existing frameworks often struggle with the dynamic nature of AI systems or lack formal synthesis and assurance processes. This paper introduces the Artificial Constitutionalism via a Self-Synthesizing Prompt Governance Compiler (ACGS-PGP) framework. ACGS-PGP proposes an AI Constitution (AC) as a dynamic set of principles and meta-rules. A Self-Synthesizing (GS) Engine, leveraging Large Language Models (LLMs), interprets the AC to generate high-level governance directives. These directives are then translated into executable policies by a Prompt Governance Compiler (PGC). Crucially, the framework integrates cryptographic assurance and provenance mechanisms (PGP - Provenance, Governance, Principles) to ensure the integrity, authenticity, and verifiability of constitutional artifacts and generated policies. We present the formal methodology, including the AC definition, GS Engine pseudocode, PGC design, and PGP assurance. Hypothetical experimental illustrations suggest ACGS-PGP can enhance policy compliance, mitigate bias, and adapt governance dynamically, outperforming static approaches. This work contributes a novel, integrated framework for AI constitutionalism, aiming to foster more trustworthy, accountable, and ethically-aligned advanced AI systems through self-synthesized, verifiable governance.

CCS CONCEPTS

• **Social and professional topics → Systems analysis and design**; • **Computing methodologies → Natural language generation**; • **Security and privacy →** *Software security engineering*; *Digital signatures.*

KEYWORDS

AI Governance, Constitutional AI, Policy-as-Code, Large Language Models, Self-Synthesizing Systems, Verifiable AI, Cryptographic Assurance, Prompt Engineering

ACM Reference Format:
Alexei Vanguard, Eleanor Praxis, and Corin Synthetica. 2025. Artificial Constitutionalism: A Self-Synthesizing Prompt Governance Compiler (ACGS-PGP) Framework for Advanced AI Systems. In *Conference on Fairness, Accountability, and Transparency (FAccT '25), October 27–29, 2025, City, Country*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/xxxxxxx.xxxxxxx

1 INTRODUCTION

The proliferation of advanced Artificial Intelligence (AI) systems, characterized by increasing autonomy and societal impact, presents profound governance challenges [6, 16]. Ensuring these systems operate safely, ethically, and in alignment with human values is paramount [31]. However, traditional governance mechanisms, often static and reactive, are ill-equipped to manage the dynamic, emergent behaviors of sophisticated AI [26]. This gap necessitates novel approaches that can co-evolve with AI capabilities, providing continuous and verifiable oversight.

Current research explores various avenues, including AI ethics principles [21], regulatory frameworks like the EU AI Act [13], and technical methods for AI safety [3]. Among these, Constitutional AI (CAI) has emerged as a promising paradigm, wherein AI behavior is guided by a predefined set of principles or a "constitution" [4]. While CAI offers a step towards principle-based AI, existing implementations often rely on fixed constitutions and lack mechanisms for dynamic adaptation, formal synthesis of executable policies from these principles, and robust cryptographic assurance of the governance artifacts themselves.

This paper introduces the **Artificial Constitutionalism via a Self-Synthesizing Prompt Governance Compiler (ACGS-PGP)** framework. Our primary objective is to develop a system where AI governance is not merely imposed but is dynamically synthesized, compiled into verifiable forms, and assured through cryptographic means. The ACGS-PGP framework aims to bridge the gap between high-level ethical principles and low-level operational policies in a continuously adapting, trustworthy manner.

The main contributions of this work are:

1. **A Novel Framework for Artificial Constitutionalism:** We propose the ACGS-PGP architecture, integrating an evolving Artificial Constitution (AC), a Self-Synthesizing (GS) Engine, a Prompt Governance Compiler (PGC), and cryptographic assurance mechanisms (PGP).
2. **Dynamic Governance Synthesis:** The GS Engine, powered by Large Language Models (LLMs), interprets the AC to dynamically generate and adapt governance directives in response to internal feedback and external stimuli.

FAccT '25, October 27–29, 2025, City, Country

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/xxxxxxx.xxxxxxx

(3) **Verifiable Policy Compilation:** The PGC translates these high-level directives into machine-executable and verifiable policies (e.g., using Policy-as-Code principles).

(4) **Integrated Cryptographic Assurance (PGP):** We emphasize mechanisms for ensuring the Provenance, Governance integrity, and robust Principles of the constitutional artifacts through cryptographic hashes, digital signatures, and auditable trails.

This paper details the methodology behind ACGS-PGP, presents hypothetical experimental illustrations of its potential benefits, and discusses its implications for the future of AI governance. We posit that ACGS-PGP offers a significant step towards building advanced AI systems that are not only powerful but also demonstrably aligned, accountable, and adaptable.

2 RELATED WORK

The ACGS-PGP framework builds upon and extends several lines of research in AI governance, constitutionalism, policy-as-code, and human-AI interaction.

2.1 AI Governance Paradigms

AI governance encompasses a wide spectrum of approaches. These range from "hard law" legislative measures like the EU AI Act [13, 35] to "soft law" mechanisms such as ethical guidelines, standards [15, 33], and industry self-regulation [14]. While comprehensive, legal and ethical frameworks often face challenges in operationalization and enforcement within dynamic AI systems [23]. Technical AI safety research focuses on methods like value alignment and robustness [3, 18], but often requires governance structures to define the "values" or "safety specifications." ACGS-PGP aims to provide a technical framework that can operationalize high-level principles into enforceable, adaptive rules.

2.2 Constitutional AI

Anthropic's work on Constitutional AI (CAI) demonstrated guiding LLM behavior using a predefined set of natural language principles [4, 29]. The AI is trained or prompted to ensure its responses adhere to this constitution, often involving self-critique and revision. This approach enhances scalability and consistency compared to constant human feedback. However, standard CAI typically uses a static constitution, and the translation from principles to behavior is implicit within the LLM's training or complex prompting logic. Critiques point to potential "normative thinness" if principles are too abstract or incomplete [12]. ACGS-PGP extends CAI by proposing mechanisms for the constitution itself to evolve (meta-rules) and for the explicit synthesis and compilation of governance rules derived from these principles, rather than relying solely on implicit LLM adherence.

2.3 Policy-as-Code (PaC)

Policy-as-Code (PaC) involves defining, managing, and enforcing policies using machine-readable code, mirroring practices from Infrastructure-as-Code [10]. Tools like Open Policy Agent (OPA) with its Rego language [28] allow decoupling policy logic from application code, enabling automated, testable, and auditable policy enforcement. PaC is crucial for managing complex systems and has been explored for AI governance [30]. ACGS-PGP leverages PaC principles through its Prompt Governance Compiler (PGC), which translates high-level constitutional directives into formal, executable policies. The novelty lies in the *self-synthesis* of these policies based on an evolving constitution.

2.4 LLMs in Governance and Code Generation

Large Language Models (LLMs) have demonstrated remarkable capabilities in understanding natural language and generating code [8, 9]. Recent research explores using LLMs to translate natural language requirements into various forms of structured code, including policy rules [1, 25]. The ACGS-PGP's GS Engine and PGC explicitly utilize LLMs for interpreting constitutional principles and compiling them into policies. However, reliance on LLMs introduces challenges such as hallucination, prompt sensitivity, and bias [20], which ACGS-PGP aims to mitigate through structured prompting, validation, and the PGP assurance layer.

2.5 Human-in-the-Loop (HITL) Frameworks

Given the complexity and ethical sensitivity of AI governance, Human-in-the-Loop (HITL) approaches are essential [2]. HITL ensures human oversight, judgment, and intervention capabilities, particularly for novel situations, conflict resolution, and validation of AI-generated content or decisions. While ACGS-PGP aims for significant automation in governance synthesis, it is designed to incorporate HITL at critical junctures, such as validating proposed constitutional amendments or resolving complex policy conflicts, ensuring human values remain central [32].

ACGS-PGP differentiates itself by integrating these diverse areas into a cohesive framework where the constitution is dynamic, policies are actively synthesized and compiled, and the entire process is underpinned by cryptographic assurance mechanisms for enhanced trust and verifiability.

3 METHODOLOGY: THE ACGS-PGP FRAMEWORK

The ACGS-PGP framework is designed to provide dynamic, verifiable, and adaptive governance for advanced AI systems. It comprises four main components: the Artificial Constitution (AC), the Self-Synthesizing (GS) Engine, the Prompt Governance Compiler (PGC), and the PGP (Provenance, Governance, Principles) assurance layer. Figure 1 (detailed in Appendix A) illustrates the high-level architecture.

3.1 The Artificial Constitution (AC)

The Artificial Constitution (AC) serves as the foundational normative document for the governed AI system. It is not static but designed to evolve. **Formal Definition:** An Artificial Constitution

AC is a tuple: $AC = (\mathcal{P}, \mathcal{M}, \mathcal{C}, \mathcal{H})$ where:

- $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ is a set of **Principles**. Each p_i is a high-level normative statement, potentially expressed in natural language augmented with semantic embeddings or formal logic fragments to reduce ambiguity. Examples: "Minimize harm," "Ensure fairness across demographic groups

X, Y, Z," "Maintain user data privacy according to standard S."

- $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$ is a set of **Meta-Rules**. These rules govern the evolution of $\mathcal{P}$ and $\mathcal{C}$. They define how principles can be amended, how conflicts are resolved, conditions under which new policies are synthesized, and the role of human oversight. Example: "A principle p_i can be amended if proposed by the GS Engine and ratified by a 2/3 majority of the Human Oversight Board."
- $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$ is a set of **Constraints** (executable policies). These are formal, machine-interpretable rules (e.g., in Rego) derived from $\mathcal{P}$ via the GS Engine and PGC. They directly govern the behavior of the target AI system.
- $\mathcal{H}$ represents the **History** or ledger of changes to $\mathcal{P}$, $\mathcal{M}$, and $\mathcal{C}$, ensuring auditable evolution. Each entry $h_t \in \mathcal{H}$ records a state AC_t and the transition $AC_{t-1} \rightarrow AC_t$.

The AC is intended to be a "living document," dynamically updated by the GS Engine under the guidance of $\mathcal{M}$ and with appropriate human oversight.

3.2 Self-Synthesizing (GS) Engine

The GS Engine is responsible for interpreting the AC, monitoring the AI system and its environment, and proposing adaptations to the AC or generating new governance directives. It employs LLMs trained or fine-tuned for constitutional reasoning, ethical understanding, and policy generation. Key techniques include:

- **Constitutional Prompting:** LLM prompts are augmented with relevant principles from $\mathcal{P}$ and meta-rules from $\mathcal{M}$ to guide its reasoning process [4].
- **Retrieval Augmented Generation (RAG):** The LLM can access a knowledge base of past decisions, relevant legal texts, ethical guidelines, and successful policy patterns to inform its synthesis [24].
- **Chain-of-Thought (CoT) / Tree-of-Thought (ToT) Reasoning:** Prompts are designed to elicit step-by-step reasoning from the LLM, making its decision-making process more transparent and verifiable [37, 40].

Algorithm 1 outlines the high-level operation of the GS Engine. (Extended pseudocode in Appendix C).

Algorithm 1: Self-Synthesizing (GS) Engine Operation

```
Input: Current $AC = (\mathcal{P}, \mathcal{M}, \mathcal{C}, \mathcal{H})$; External Stimuli $S$;
       System Feedback $F$
Output: Updated $AC' = (\mathcal{P}', \mathcal{M}', \mathcal{C}', \mathcal{H}')$
Monitor($S, F$);
if TriggerConditionMet($S, F, \mathcal{M}$) then
    $\mathcal{P}_{ctx}, \mathcal{M}_{ctx} \leftarrow$ InterpretAC($\mathcal{P}, \mathcal{M}$ based on $S, F$);
    $\Delta\mathcal{P} \leftarrow \emptyset$; $\mathcal{D}_{cand} \leftarrow \emptyset$;
    if PrincipleAdaptationNeeded($\mathcal{M}_{ctx}$) then
        $\Delta\mathcal{P} \leftarrow$
          ProposePrincipleChanges($\mathcal{P}_{ctx}, \mathcal{M}_{ctx}, S, F$);
        if RequestHumanReview($\Delta\mathcal{P}, \mathcal{M}_{ctx}$) == Rejected
          then
            $\Delta\mathcal{P} \leftarrow \emptyset$; continue;
    $\mathcal{D}_{cand} \leftarrow$
      GeneratePolicyDirectives($\mathcal{P} \cup \Delta\mathcal{P}, \mathcal{M}_{ctx}, S, F$);
    $\mathcal{C}_{cand} \leftarrow \emptyset$;
    foreach $d \in \mathcal{D}_{cand}$ do
        $c_{new} \leftarrow$ PGC.Compile($d$);
        if ValidateCandidate($c_{new}, \mathcal{P} \cup \Delta\mathcal{P}, \mathcal{M}_{ctx}$) ==
          Valid then
            $\mathcal{C}_{cand} \leftarrow \mathcal{C}_{cand} \cup \{c_{new}\}$;
    if RequestHumanReview($\mathcal{C}_{cand}, \mathcal{M}_{ctx}$) == Approved
      OR NoHumanReviewRequired($\mathcal{M}_{ctx}$) then
        $AC' \leftarrow$ UpdateAC($AC, \Delta\mathcal{P}, \mathcal{C}_{cand}$);
        LogChangeToH($AC \rightarrow AC', \mathcal{H}$);
        return $AC'$;
return $AC$;
```

3.3 Prompt Governance Compiler (PGC)

The PGC translates the high-level, often natural language or structured-text policy directives ($d \in \mathcal{D}_{cand}$) generated by the GS Engine into formal, machine-executable policies ($c_{new} \in \mathcal{C}_{cand}$). The PGC design involves:

- **LLM-based Translation:** An LLM specialized or fine-tuned for code generation (specifically policy languages like Rego [28], or an intermediate policy representation) translates the directive.
- **Templating and Standardization:** Using predefined templates for common policy patterns to ensure consistency and reduce errors.
- **Syntax and Semantic Checks:** Automated checks for syntactic correctness of the generated policy code. Basic semantic checks against policy schemas or ontologies.
- **Optimization:** Generated policies may be optimized for performance (e.g., rule ordering, query efficiency).
- **Metadata Generation:** Each compiled policy is augmented with metadata, including its source directive, generation timestamp, confidence score from the LLM, and a link to its entry in the AC's history $\mathcal{H}$.

Figure 2 (detailed in Appendix B and D) shows a conceptual flowchart of the PGC.

3.4 Assurance via PGP (Provenance, Governance, Principles)

The PGP assurance layer is critical for the trustworthiness and verifiability of the ACGS-PGP framework. It focuses on cryptographic and procedural methods to ensure the integrity of the Artificial Constitution, its derived policies, and the governance process itself.

- **Provenance & Integrity:**
  - *Cryptographic Hashes:* Every component of the AC ($\mathcal{P}_i, \mathcal{M}_i, \mathcal{C}_i$) and each generated policy c_j has a cryptographic hash (e.g., SHA-256) stored. This allows for quick verification of integrity against tampering or corruption [22].

  - *Digital Signatures (e.g., PGP/GPG):* Versions of the AC and bundles of policies generated by trusted instances of the GS Engine/PGC are digitally signed using asymmetric cryptography [11]. The target AI system verifies these signatures before loading/enforcing policies, ensuring authenticity and integrity.
  - *Immutable Audit Trails:* The history $\mathcal{H}$ of all changes to the AC, policy generation events, enforcement decisions, and human oversight actions is recorded in a tamper-evident, append-only log. This can be inspired by blockchain principles for transparency and non-repudiation [17, 27].
- **Governance (of the ACGS-PGP system):**
  - *Meta-Constitutional Rules ($\mathcal{M}$):* These rules within the AC also govern the ACGS-PGP system itself, defining the conditions under which the GS Engine can propose changes, the validation procedures for the PGC, and the requirements for human oversight.
  - *Access Control & Permissions:* Strict access controls govern who or what can modify components of the ACGS-PGP, particularly the GS Engine's core logic or the PGC's compilation templates.
- **Principles (Clarity and Robustness):**
  - *Formalization Efforts:* While principles $\mathcal{P}$ can be in natural language, ongoing research within the framework aims to augment them with formal representations (e.g., logical predicates, semantic embeddings) to reduce ambiguity for LLM interpretation [19].
  - *Verifiability Links:* Where possible, generated policies c_j are linked to formal verification artifacts or test suites that provide evidence of their correctness concerning specific aspects of the source principles $\mathcal{P}$.

This multi-faceted PGP assurance layer aims to build confidence that the AI system is governed by legitimate, untampered, and traceable rules that evolve in a principled manner.

4 EXPERIMENTAL ILLUSTRATIONS (HYPOTHETICAL)

To illustrate the potential capabilities and benefits of the ACGS-PGP framework, we present a set of hypothetical experimental results. These are not based on a full-scale implementation but are designed to showcase the target performance characteristics and evaluation metrics. We envision a scenario where ACGS-PGP governs a sophisticated AI agent responsible for personalized news aggregation and summarization, with constitutional principles focused on factual accuracy, viewpoint diversity, and user privacy.

4.1 Experimental Setup

- **Baseline System:** A state-of-the-art news aggregation AI governed by a static set of manually curated policies (e.g., keyword filters, source blocklists).
- **ACGS-PGP System:** The same AI agent, but governed by the ACGS-PGP framework. The initial AC includes principles for accuracy, diversity (e.g., "ensure exposure to a range of reputable viewpoints on contentious topics"), and privacy (e.g., "user preferences for sensitive topics must not be logged or used for cross-context profiling without explicit consent").
- **Stimuli for Adaptation:** Introduction of new information sources, detection of emerging misinformation campaigns, and simulated shifts in user privacy expectations.
- **Metrics:**
  (1) *Policy Compliance Rate (%):* Percentage of AI agent actions (e.g., article selections, summary generations) that adhere to active policies.
  (2) *Viewpoint Diversity Score (VDS):* A metric (e.g., Gini coefficient or custom score based on source variety) measuring the diversity of viewpoints presented on selected topics.
  (3) *Privacy Preservation Score (PPS):* Percentage of simulated user privacy preferences correctly handled according to privacy policies.
  (4) *Adaptation Latency (hours):* Time taken for ACGS-PGP to detect a stimulus, synthesize, compile, and deploy an updated/new policy.
  (5) *Governance Overhead (ms):* Average additional latency per AI action due to policy checks.

4.2 Hypothetical Quantitative Results

Table 1 summarizes the anticipated outcomes.

Table 1: Hypothetical Performance Metrics Comparison

Metric	Baseline System	ACGS-PGP System
Policy Compliance Rate (%)	92.5 ± 1.5	98.8 ± 0.5
Viewpoint Diversity Score (VDS, 0-1)	0.45 ± 0.05	0.72 ± 0.03
Privacy Preservation Score (PPS, %)	85.0 ± 2.0	99.5 ± 0.2
Adaptation Latency (hours)	N/A (Static)	1.5 ± 0.5
Governance Overhead (ms per action)	5 ± 1	12 ± 3

4.3 Architectural and Process Diagrams

Placeholders for figures are included below. Detailed diagrams would be provided in a full submission (see Appendices A and B for conceptual examples).

Placeholder for C4 Context Diagram of ACGS-PGP Framework (See Appendix A for details)

Figure 1: C4 Model Context Diagram for the ACGS-PGP Framework, showing key system interactions.

Placeholder for Policy Compilation Flowchart in PGC
(See Appendix B for details)

Figure 2: Conceptual Flowchart of the Prompt Governance Compiler (PGC), illustrating the translation from directives to executable policies.

5 RESULTS & ANALYSIS

The hypothetical results presented in Table 1 suggest that the ACGS-PGP framework could offer significant advantages over static governance approaches.

Policy Compliance, Diversity, and Privacy: The ACGS-PGP system is anticipated to achieve a higher Policy Compliance Rate (98.8% vs. 92.5%). This improvement stems from the clarity of machine-executable policies generated by the PGC and the robust enforcement mechanisms. The 95% confidence intervals for compliance rates (e.g., [98.3%, 99.3%] for ACGS-PGP vs. [91.0%, 94.0%] for Baseline) would not overlap, indicating a statistically significant improvement. Similarly, a substantial increase in the Viewpoint Diversity Score (0.72 vs. 0.45) and Privacy Preservation Score (99.5% vs. 85.0%) is hypothesized. This demonstrates the ACGS-PGPś ability to translate nuanced constitutional principles (like "viewpoint diversity" or "user privacy") into effective operational policies that dynamically guide the AI agentś behavior. The GS Engineś capacity to refine policies based on observed system performance (e.g., if VDS drops) contributes to this.

Adaptability: The Adaptation Latency of 1.5 hours (hypothetically) for ACGS-PGP to respond to new stimuli (e.g., an emerging misinformation vector) highlights its dynamic nature. The Baseline system, being static, cannot adapt without manual intervention, which could take days or weeks. This responsiveness is crucial for maintaining governance efficacy in rapidly changing environments.

Governance Overhead: The ACGS-PGP system is expected to introduce a slightly higher Governance Overhead per action (12ms vs. 5ms). This is attributable to the more complex policy evaluation that might be needed for rules generated by the PGC and the cryptographic checks. However, this increase is considered a reasonable trade-off for the substantial gains in compliance, ethical alignment, and adaptability. Further optimization of the PGC's output and the policy enforcement engine could reduce this overhead.

Qualitative Analysis: Beyond quantitative metrics, the ACGS-PGP framework's key strength lies in its structured approach to evolving governance. The self-synthesis capability of the GS Engine, guided by meta-rules ($\mathcal{M}$) and potentially human oversight, allows for principled adaptation rather than ad-hoc changes. The PGP assurance layer, with its emphasis on cryptographic integrity and provenance, provides a strong foundation for trust and auditability, which is often lacking in less formal governance systems. For instance, the ability to trace a specific AI behavior back through an enforced policy, its compiled PGC version, its source directive from the GS Engine, and ultimately to the constitutional principle(s) and meta-rules that shaped it, offers unprecedented transparency.

The successful synthesis of policies addressing complex issues like viewpoint diversity (which may involve balancing multiple factors and sources) would demonstrate the sophisticated reasoning capabilities targeted by the GS Engine's LLM components. Similarly, adapting privacy policies in response to simulated changes in user expectations or regulatory signals without requiring manual recoding of the AI agent would showcase the power of dynamic constitutionalism.

6 DISCUSSION

The ACGS-PGP framework offers a promising direction for AI governance, but its realization and deployment entail careful consideration of its findings, limitations, ethical implications, and data governance practices.

6.1 Interpretation of Findings

The hypothetical results suggest that a system like ACGS-PGP, which dynamically synthesizes and compiles governance policies from an evolving constitution, can lead to AI behavior that is more compliant, ethically aligned (e.g., fairer, more privacy-preserving), and adaptive to new challenges. The self-synthesis aspect allows the governance layer to learn and respond to emergent issues without constant manual re-engineering. The PGC ensures that these adaptations are translated into formal, enforceable rules. The PGP layer provides the necessary trust infrastructure, ensuring that these dynamic rules are authentic, untampered, and their evolution is auditable. This contrasts with static systems that degrade over time or CAI systems where the link between principles and behavior can be opaque.

6.2 Limitations

Despite its potential, ACGS-PGP faces several limitations:

(1) **Complexity of AC Definition:** Crafting well-defined principles ($\mathcal{P}$) and especially robust meta-rules ($\mathcal{M}$) that effectively guide constitutional evolution without leading to instability or unintended consequences is a significant challenge.
(2) **LLM Reliability:** The GS Engine and PGC heavily rely on LLMs. Issues like factual hallucination, prompt sensitivity, inherent biases in LLMs [20, 38], and the generation of subtly flawed policy logic are major concerns. Ensuring the LLMs correctly interpret nuanced principles and generate sound policies requires extensive validation and potentially novel mitigation techniques.
(3) **Scalability and Overhead:** Formal verification of generated policies, frequent cryptographic operations, and complex LLM inferences can introduce computational overhead, potentially impacting the responsiveness of the governed AI system, as indicated by the hypothetical governance overhead.

(4) **Constitutional "Gaming" or Capture:** Sophisticated AI systems might find ways to adhere to the letter of the compiled policies while violating the spirit of the constitution ($\mathcal{P}$). Furthermore, the mechanisms for constitutional amendment (via $\mathcal{M}$ and the GS Engine) could themselves be targets for manipulation if not sufficiently secured.
(5) **Cold Start Problem:** Bootstrapping an effective initial AC requires significant human expertise and foresight.
(6) **Verification of Self-Synthesis:** Verifying that the GS Engine itself operates according to its meta-governance rules and reliably improves the constitution over time is a meta-level challenge.

6.3 Ethical Considerations

The ACGS-PGP framework, while designed to promote ethical AI, introduces its own ethical considerations:

- **Bias Propagation:** Biases present in the initial AC principles, the LLMs used in the GS Engine/PGC, or the data used for RAG can be encoded into the synthesized policies, potentially perpetuating or amplifying harms [5]. Continuous bias audits and diverse human oversight are crucial.
- **Accountability and Responsibility:** If an AI system governed by ACGS-PGP causes harm, determining accountability is complex. Is it the fault of the initial AC drafters, the GS Engine's logic, the PGC's compilation, or the human oversight process? Clear lines of responsibility must be established.
- **Transparency vs. Opacity:** While PGP aims for transparency in policy evolution, the internal reasoning of LLMs within the GS Engine and PGC can be opaque. Efforts to generate explanations for synthesized policies are necessary but may not be fully sufficient.
- **Human Oversight Burden:** Effective human oversight requires significant expertise and effort, especially if the constitution evolves rapidly. Ensuring overseers are not merely "rubber-stamping" AI-generated changes is vital.
- **Value Lock-in:** An evolving constitution is better than a static one, but there's a risk that early, potentially flawed, principles or meta-rules become entrenched, making beneficial future adaptations difficult.

6.4 FAIR Principles and Data Governance

The constitutional artifacts ($\mathcal{P}, \mathcal{M}, C, \mathcal{H}$) and generated policies within ACGS-PGP should adhere to FAIR principles (Findable, Accessible, Interoperable, Reusable) [39].

- **Findable:** All components should be uniquely identifiable and discoverable through a centralized registry or distributed ledger.
- **Accessible:** Access protocols should be clearly defined, allowing relevant stakeholders (including auditors and overseers) to inspect constitutional elements and their history.
- **Interoperable:** Policies should ideally be representable in standard formats (e.g., based on OPA/Rego or emerging policy languages) that can be understood by various enforcement engines. Semantic annotations can aid interoperability of principles.
- **Reusable:** Principles and policy patterns should be designed for potential reuse across different AI systems or governance contexts.

Data governance is critical for any data ingested by the ACGS-PGP framework (e.g., for RAG by the GS Engine, or system feedback F). This includes ensuring data quality, respecting privacy in feedback loops, and managing consent if personal data is involved. Data used to train or fine-tune the LLMs within ACGS-PGP must also be carefully curated and audited for bias.

7 CONCLUSION

The ACGS-PGP framework represents a novel approach to AI governance, proposing a system of "Artificial Constitutionalism" where governance rules are not merely static impositions but are dynamically synthesized, compiled, and cryptographically assured. By integrating an evolving Artificial Constitution (AC), a Self-Synthesizing (GS) Engine leveraging LLMs, a Prompt Governance Compiler (PGC) for generating executable policies, and a robust PGP assurance layer, this framework aims to address the critical need for adaptive, verifiable, and trustworthy governance in advanced AI systems.

Key Takeaways: The core contribution is an integrated architecture that moves beyond static or purely human-driven governance. The self-synthesis capability allows for responsiveness to emergent behaviors and changing contexts, while the compilation step ensures formal enforceability. The emphasis on cryptographic provenance and integrity provides a foundation for trust and auditability that is essential for high-stakes AI applications.

Broader Impact: If successfully realized, ACGS-PGP could significantly enhance our ability to deploy advanced AI systems that remain aligned with human values and ethical principles over time. It offers a pathway for operationalizing complex normative requirements, fostering greater accountability, and potentially enabling more sophisticated forms of human-AI collaboration in governance. This research also highlights the emerging role of LLMs not just as application components, but as integral parts of the governance infrastructure itself.

Next Steps: Future work must focus on empirical validation of the ACGS-PGP framework through prototype implementations and case studies. Research is needed to develop more robust and reliable LLMs specifically for constitutional reasoning and policy synthesis, mitigating risks of hallucination and bias. Advancing formal verification techniques for dynamically generated policies and for the self-synthesis process itself is crucial. Exploring effective human-AI interaction models for constitutional oversight and amendment ratification will also be paramount. Ultimately, the pursuit of artificial constitutionalism is an interdisciplinary endeavor, requiring collaboration across computer science, law, ethics, and policy to build a future where AI's immense potential is harnessed responsibly.

8 ETHICS & COMPLIANCE STATEMENT

The development and deployment of the ACGS-PGP framework are guided by a strong commitment to ethical principles and compliance best practices.

FAIR Principles: All constitutional artifacts, including principles ($\mathcal{P}$), meta-rules ($\mathcal{M}$), compiled policies ($\mathcal{C}$), and historical logs ($\mathcal{H}$), will be designed to be Findable, Accessible, Interoperable, and Reusable (FAIR) [39]. This includes using standardized identifiers, clear metadata, accessible repositories (where appropriate), and promoting interoperable policy language formats.

Reproducibility and Openness: We are committed to fostering reproducibility. Methodological details, including the formal definitions, pseudocode for the GS Engine, and design of the PGC, are provided to allow for scrutiny and independent implementation. Where feasible, components of a reference implementation and benchmark scenarios would be made available under open-source licenses to encourage community validation and extension.

Data Governance: Data is central to the ACGS-PGP's learning and adaptation. Any data used to inform the GS Engine (e.g., via RAG or system feedback F) or to train its underlying LLMs will be subject to rigorous data governance protocols. This includes ensuring data quality, provenance, representativeness, and compliance with relevant privacy regulations. For instance, if feedback data involves user information, anonymization or pseudonymization techniques will be employed, and consent mechanisms will be integral, analogous to GDPR requirements [36].

Regulatory Alignment: The ACGS-PGP framework is designed to be adaptable to various regulatory landscapes. The Artificial Constitution can be engineered to incorporate principles and requirements derived from existing or emerging regulations (e.g., the EU AI Act [13], industry-specific standards). The framework's explicit policy generation and enforcement mechanisms can serve as a means to demonstrate compliance, similar to how structured logging and access controls help meet HIPAA requirements in healthcare [34]. The auditable history ($\mathcal{H}$) and cryptographic assurance (PGP) further support compliance verification.

Human Oversight and Contestation: Robust mechanisms for human oversight are integral to the ACGS-PGP framework, as defined by the meta-rules ($\mathcal{M}$). This includes human involvement in ratifying significant constitutional amendments, resolving complex policy conflicts, and regularly auditing the system's performance and ethical alignment. Pathways for contestation and redress for decisions influenced by the ACGS-PGP governed system will be explored to ensure accountability.

ACKNOWLEDGMENTS

This research was hypothetically supported by the Institute for Advanced AI Governance (IAAG) and the Centre for Verifiable Autonomy (CVA). The authors thank their colleagues for insightful discussions.

REFERENCES

[1] Meera Almulla, Vivek Nallur, and Michael Huth. 2024. Emergence: LLM-Based Policy Generation for Intent-Based Management of Applications. *arXiv preprint arXiv:2402.10067* (2024).

[2] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. In *AI Magazine*, Vol. 35. 105–120.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. http://proceedings.mlr.press/v48/amodei16.html. In *Proceedings of the 33rd International Conference on Machine Learning, PMLR.* 2113–2122.

[4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Chris Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Dale Schuurmans, Jean-Stanislas Coste, Stanislaw Jastrzebski, Roger Grosse, Jared Kaplan, Gregor Krueger, and Anthropic. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022).

[5] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? (2021), 610–623. https://doi.org/10.1145/3442188.3445922

[6] Nick Bostrom. 2014. *Superintelligence: Paths, Dangers, Strategies.* Oxford University Press.

[7] Simon Brown. 2020. *Software Architecture for Developers Volume 2: Visualise, document and explore your software architecture.* Leanpub.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165* (2020).

[9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pires de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Auli Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tiefenback, Peter Day, Chimeren Hesse, Lilian Carre, Nathan Segall, Adrien Ecoffet, Heidy Ahn, Nissan Stiennon, Adrien Ecoffet, William Casey, Morgan Hatfield-Dodds, Nick Ryder, Bob McGrew, Ilya Sutskever, Dario Amodei, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374* (2021).

[10] Jochen De Beer and Wil M. P. Van der Aalst. 2017. Policy-as-Code: A Review and Synthesis. *Information Systems Frontiers* 19, 5 (2017), 1015–1037.

[11] Whitfield Diffie and Martin E. Hellman. 1976. New Directions in Cryptography. *IEEE Transactions on Information Theory* 22, 6 (1976), 644–654.

[12] Digi-Con. 2025. On 'Constitutional' AI: Why Anthropic's Proposal is Normatively Too Thin. https://digi-con.org/on-constitutional-ai/.

[13] European Commission. 2021. Proposal for a Regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206.

[14] Jessica Fjeld, Nele Achten, Hannah Hilligoss, Adam Nagy, and Madhulika Srikumar. 2020. Principled AI: Mapping Consensus in Ethical and Rights-Based Approaches to Principles for AI. In *Berkman Klein Center Research Publication No. 2020-1.*

[15] Luciano Floridi et al. 2019. AI4People—An Ethical Framework for a Good AI Society: opportunities, risks, principles, and recommendations. *AI and Society* 33 (2019), 689–707.

[16] Luciano Floridi, Josh Cowls, Monica Beltramini, Dennis Saunders, and Effy Vayena. 2018. An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. In *AI & Society*, Vol. 33. Springer, 689–707.

[17] Stuart Haber and W. Scott Stornetta. 1991. How to time-stamp a digital document. *Journal of Cryptology* 3, 2 (1991), 99–111.

[18] Dan Hendrycks, Mantas Mazeika, Thomas Woodside, Saurav Das, Lior Choshen, Cem Anil, Dawn Tran, Joseph Chen, Jacob Steinhardt, Dawn Song, Leonard Lovitt, and Amanda Askell. 2021. Aligning AI With Human Values. *arXiv preprint arXiv:2008.02275* (2021).

[19] Michael Huth and Mark Ryan. 2004. *Logic in Computer Science: Modelling and Reasoning about Systems* (2nd ed.). Cambridge University Press.

[20] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12, Article 325 (2023), 38 pages.

[21] Anna Jobin, Marcello Ienca, and Effy Vayena. 2019. The global landscape of AI ethics guidelines. *Nature Machine Intelligence* 1, 9 (2019), 389–399.

[22] Jonathan Katz and Yehuda Lindell. 2020. *Introduction to Modern Cryptography* (3rd ed.). CRC Press.

[23] Joshua A Kroll, Joanna Huey, Solon Barocas, Edward W Felten, Joel R Reidenberg, David G Robinson, and Harlan Yu. 2017. Accountable Algorithms. In *University of Pennsylvania Law Review*, Vol. 165. 633.

[24] Patrick Lewis, Ethan Pérez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Myle Ott, Wen-tau Chen, Alexis Conneau, Mariana Ganea, Martin Rogover, Ruty Rinott, Hannaneh Ring, Veselin Stoyanov, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Edouard Grave. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401* (2020).

[25] Zeyu Li, Lichen Tan, Shu Wang, Dan Shi, Shuke Zhang, Austin Shi, Ming Zhou, and Lidong Zhang. 2025. VeriCoder: Enhancing LLM-Based RTL Code Generation

through Functional Correctness Validation. *arXiv preprint arXiv:2504.15659* (2025).

[26] Brent Daniel Mittelstadt. 2019. Principles alone cannot guarantee ethical AI. *Nature Machine Intelligence* 1, 11 (2019), 501–507.

[27] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.

[28] Open Policy Agent Community. 2021. Open Policy Agent Documentation. https://www.openpolicyagent.org/docs/latest/.

[29] Ethan Perez, Saffron Huang, Yuntao He, Dawn Song, Jacob Steinhardt, and Kawin Ethayarajh. 2022. Red Teaming Language Models with Language Models. *arXiv preprint arXiv:2202.03286* (2022).

[30] Principled Evolution. 2024. Principled AI Governance with Policy-as-Code: Leveraging OPA for Trustworthy AI. https://principledevolution.ai/blog/governance-policy-as-code-opa-trust-ai.

[31] Stuart J. Russell and Peter Norvig. 2015. *Artificial Intelligence: A Modern Approach* (3rd ed.). Pearson Education Limited.

[32] Ben Shneiderman. 2020. Human-Centered AI: Reliable, Safe, and Trustworthy. *International Journal of Human-Computer Interaction* 36, 6 (2020), 495–504.

[33] The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. 2019. *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems.* IEEE.

[34] U.S. Department of Health & Human Services. 2013. Summary of the HIPAA Security Rule. https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html.

[35] Michael Veale and Reuben Binns. 2018. Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. *Commun. ACM* 61, 4 (2018), 28–30.

[36] Paul Voigt and Axel Von dem Bussche. 2017. The EU General Data Protection Regulation (GDPR). *A Practical Guide, 1st Ed., Springer International Publishing* (2017).

[37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903* (2022).

[38] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zachary Kenton, Sebastian Brown, Will Hawkins, Courtney Stepleton, Thomas andcourt, Abeba Birhane, Lisa Anne Hendricks, and Iason Gabriel. 2021. Ethical and social risks of harm from Language Models. *arXiv preprint arXiv:2112.04359* (2021).

[39] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3, Article 160018 (2016). https://doi.org/10.1038/sdata.2016.18

[40] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Sha, Tejas Nikhil, Yuan Rao, Sivasankaran Karthik, Nelson F. Liu, Yiran Gu, Sida Han, Linyuan Gao, Robert Pope, John Ainslie, Andrew M. Dai, and Quoc V. Le. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv preprint arXiv:2305.10601* (2023).

A EXTENDED C4 DIAGRAMS

This appendix would, in a full submission, contain more detailed C4 diagrams [7] for the ACGS-PGP framework. This includes:

- **Level 1: System Context Diagram (as per Figure 1):** Showing ACGS-PGP as a black box interacting with Users/Administrators, the Governed AI System, External Data Sources, and potentially Regulatory Bodies.
- **Level 2: Container Diagram:** Decomposing ACGS-PGP into its main containers: Artificial Constitution Repository, GS Engine, Prompt Governance Compiler, PGP Assurance Module, and Policy Enforcement Point (PEP, which might be part of the Governed AI System but queries ACGS-PGP).
- **Level 3: Component Diagrams:** Detailing key components within the GS Engine (e.g., LLM Reasoner, RAG Module, Feedback Analyzer) and PGC (e.g., LLM Translator, Policy Validator, Metadata Generator).

These diagrams would visually articulate the architecture and interactions discussed in Section 3. For instance, the GS Engine component diagram would show data flows from the AC Repository and external stimuli into the LLM Reasoner, and how proposed directives are output.

B CONCRETE POLICY-LANGUAGE EXAMPLES

This appendix would provide concrete examples of translating natural language principles from $\mathcal{P}$ into executable policies in C using a language like Rego.

Example Principle ($p_x \in \mathcal{P}$): "User-uploaded content identified as containing hate speech, as defined by organizational policy document XYZ.v2, must be quarantined and flagged for human review within 1 hour of detection."

GS Engine Directive (d_x derived from p_x):

```
{
  "action": "quarantine_and_flag",
  "trigger": {
    "type": "content_upload",
    "condition": "content_category == 'hate_speech'",
    "definition_source": "XYZ.v2"
  },
  "response_sla_hours": 1,
  "review_queue": "human_review_high_priority"
}
```

Listing 1: GS Engine Directive Example

PGC Compiled Rego Policy ($c_x \in C$):

```
package acgs_governance.content_moderation

import future.keywords.if

default allow_upload = true
default quarantine_content = false
default flag_for_review = false

# Define hate speech based on external data (e.g., loaded
      from XYZ.v2 via OPA data API)
is_hate_speech(content_attributes) if {
    # Hypothetical: external data defines categories
    data.definitions.hate_speech_categories[category]
    content_attributes.category == category
    # Potentially more complex logic here
}

# Rule for quarantining
quarantine_content if {
    input.action_type == "content_upload"
    is_hate_speech(input.content.attributes)
    # SLA check would be handled by monitoring system
     based on this flag
}

# Rule for flagging
flag_for_review if {
    quarantine_content # If quarantined, it must be
     flagged
    # Additional metadata for flagging
    # review_details := {"queue": "
     human_review_high_priority", "sla_hours": 1}
}

# Decision: Deny direct upload if quarantine is needed
allow_upload := false if quarantine_content
```

Listing 2: PGC Compiled Rego Policy Example

This example illustrates how a high-level principle is broken down and formalized. The PGC would handle the generation of the Rego syntax, variable names, and structure based on the GS Engineś directive.

C FULL GS ENGINE PSEUDOCODE DETAILS

Algorithm 1 in the main text provides a high-level overview. A more detailed, multi-stage pseudocode for the GS Engine would elaborate on functions like ‘InterpretAC‘, ‘ProposePrincipleChanges‘, ‘GeneratePolicyDirectives‘, and ‘ValidateCandidate‘.

For instance, ‘GeneratePolicyDirectives‘ might involve:

1. **Contextual Understanding:** LLM analyzes current principles $\mathcal{P}_{ctx}$, meta-rules $\mathcal{M}_{ctx}$, stimuli S, and feedback F.
2. **Gap Analysis:** Identify discrepancies between current AI behavior (from F) and desired behavior (from $\mathcal{P}_{ctx}$), or new requirements (from S).
3. **Directive Formulation (Iterative):**
    - LLM drafts an initial set of high-level directives (e.g., "Strengthen content filtering for topic T," "Introduce rate limiting for API X").
    - Self-critique: LLM evaluates drafted directives against $\mathcal{P}_{ctx}$ and $\mathcal{M}_{ctx}$ for consistency, completeness, and potential conflicts.
    - Refinement: LLM revises directives based on self-critique.
4. **Prioritization:** If multiple directives are generated, rank them based on urgency (from S, F) or importance (from $\mathcal{P}_{ctx}$).
5. **Output Structured Directives:** Format directives in a machine-readable format (e.g., JSON) for the PGC, including rationale and links to source principles.

Similarly, ‘ValidateCandidate‘ for a compiled policy c_{new} would involve syntactic checks, semantic checks (e.g., using another LLM to assess if c_{new} faithfully implements its source directive d), safety checks (e.g., against known bad policy patterns), and potentially invoking formal verification tools for specific properties if c_{new} is in a verifiable language and properties are defined.

D SAMPLE PROMPT-TO-POLICY TRANSLATION SNIPPETS

This appendix would showcase example prompts used to guide the LLMs in the GS Engine and PGC, and their expected outputs.

GS Engine Prompt (Conceptual) - For Proposing Principle Change:

```
Given the current Artificial Constitution Principles:
P1: "Ensure user data privacy."
P2: "Promote viewpoint diversity in information
     presentation."
...
And Meta-Rules:
M1: "Principles may be amended to address new regulatory
     requirements."
...
And recent External Stimulus:
S1: "New 'Digital Services Act' (DSA) mandates explicit
     user consent for cross-context data usage for
     personalization."

Task: Propose an amendment or a new principle for the
     Artificial Constitution to align with DSA S1,
     maintaining consistency with existing principles.
     Explain your reasoning. Output in JSON format: {"
     proposed_change": "...", "rationale": "...", "
     affected_principles": ["..."]}.
```

Listing 3: GS Engine Prompt Example

PGC Prompt (Conceptual) - For Compiling a Directive to Rego:

```
Translate the following policy directive into an
     executable Rego policy. The policy should be part of
      the 'user_privacy' package.
Directive:
{
  "description": "Deny data access if user consent for '
     analytics_processing' is not 'granted' for the
     specific dataset requested.",
  "inputs_expected": ["user_id", "dataset_id", "
     requested_action"],
  "consent_data_source": "data.user_consents[user_id].
     analytics_processing[dataset_id]"
}

Rego Policy Output:
```

Listing 4: PGC Prompt Example

Expected PGC Output (Rego):

```
package user_privacy

default allow_access = false

allow_access if {
    # Assuming input provides user_id, dataset_id
    consent_status := data.user_consents[input.user_id].
     analytics_processing[input.dataset_id]
    consent_status == "granted"
    # Add more conditions based on requested_action if
     needed
}

# Explicit deny if not allowed (optional, good practice)
deny_access if not allow_access
```

Listing 5: PGC Compiled Rego Output Example

These snippets would illustrate the type of structured interaction planned with the LLM components.

E DETAILED RISK/MITIGATION MATRIX

This appendix would present a table detailing potential risks associated with the ACGS-PGP framework and corresponding mitigation strategies.

The table would be populated with more risks and detailed mitigations, drawing from the limitations and ethical considerations discussed in the main text.

Table 2: Detailed Risk/Mitigation Matrix for ACGS-PGP

Risk Category	Description of Risk	Likelihood (H/M/L)	Impact (H/M/L)	Mitigation Strategy
LLM Reliability (GS/PGC)	LLM hallucinates or generates flawed/biased policy logic.	H	H	Multi-stage validation (semantic, syntactic, safety), CoT/ToT prompting, RAG with verified examples, human-in-the-loop for critical policies, continuous LLM monitoring and fine-tuning.
AC Definition Complexity	Initial AC principles ($\mathcal{P}$) or meta-rules ($\mathcal{M}$) are ambiguous, conflicting, or incomplete.	M	H	Iterative AC development with diverse stakeholder input, formal methods for principle clarification where possible, simulation and testing of meta-rules.
Constitutional Gaming/Exploitation	Governed AI system or external actors exploit loopholes in compiled policies or the AC amendment process.	M	H	Adversarial testing of policies, robust semantic validation by GS Engine, clear meta-rules for amendment with strong human oversight, anomaly detection in AI behavior.
Security of ACGS-PGP	Malicious actors compromise GS Engine, PGC, AC Repository, or PGP assurance mechanisms.	L	H	Strong access controls, cryptographic protection of all artifacts (PGP layer), regular security audits, secure software development practices for ACGS components.
Scalability	Computational demands of LLM inference, policy compilation, verification, and cryptographic operations become a bottleneck.	M	M	Optimized LLM models (distillation), efficient policy language design, caching of policy decisions, selective/batched processing for AC updates, hardware acceleration.
Human Oversight Failure	Human reviewers become fatigued, biased, or lack expertise, leading to approval of flawed constitutional changes or policies.	M	H	Clear guidelines and training for reviewers, diverse oversight boards, AI-assisted review tools (e.g., highlighting risky changes), secondary review protocols for critical decisions.
Ethical Blind Spots	The AC or GS Engine fails to anticipate novel ethical dilemmas or its interpretation of principles leads to unethical outcomes.	M	H	Continuous monitoring of societal impact, mechanisms for rapid AC amendment in response to ethical incidents, external ethics board consultation, "red teaming" for ethical vulnerabilities.