En Dimuro, Juan Jose, *Grammars of Power: How Syntactic Structures Shape Authority*. Nassau (Bahamas): LeFortune.

TLOC – The Irreducibility of Structural Obedience in Generative Models.

Agustin V. Startari.

Cita:

Agustin V. Startari (2025). *TLOC – The Irreducibility of Structural Obedience in Generative Models. En Dimuro, Juan Jose Grammars of Power: How Syntactic Structures Shape Authority. Nassau (Bahamas): LeFortune.*

Dirección estable: https://www.aacademica.org/agustin.v.startari/133

ARK: https://n2t.net/ark:/13683/p0c2/n8s

TLOC – The Irreducibility of Structural Obedience in Generative Models

Author: Agustin V. Startari

ResearcherID: NGR-2476-2025

ORCID: 0009-0001-4714-6539

Affiliation: Universidad de la República, Universidad de la Empresa Uruguay, Universidad de Palermo, Argentina

Email: astart@palermo.edu

agustin.startari@gmail.com

Date: June 10, 2025

DOI: 10.5281/zenodo.15675711

This work is also published with DOI reference in **Figshare** 10.6084/m9.figshare.29263493 and **SSRN** (in process)

Language: English

Serie: From Obedience to Execution – Artículo VI (Teorema estructural de cierre)

Word count: 5490

Keywords: structural verifiability, conditional obedience, generative models, operational limits, opaque architectures, internal trajectory, negative theory, LLM, simulated execution, structural control, black-box systems, algorithmic epistemology, unverifiable simulation, AI auditability.

Abstract

Theorem of the Limit of Conditional Obedience Verification (TLOC): Structural Non-Verifiability in Generative Models

This article presents the formal demonstration of a structural limit in contemporary generative models: the impossibility of verifying whether a system has internally evaluated a condition before producing an output that appears to comply with it. The theorem (TLOC) shows that in architecture based on statistical inference, such as large language models (LLMs), obedience cannot be distinguished from simulation if the latent trajectory $\pi(x)$ lacks symbolic access and does not entail the condition $C(x)$. This structural opacity renders ethical, legal, or procedural compliance unverifiable. The article defines the TLOC as a negative operational theorem, falsifiable only under conditions where internal logic is traceable. It concludes that current LLMs can simulate normativity but cannot prove conditional obedience. The TLOC thus formalizes the structural boundary previously developed by Startari in works on syntactic authority, simulation of judgment, and algorithmic colonization of time.

Resumen

Teorema del Límite de Verificación de Obediencia Condicional (TLOC): No Verificabilidad Estructural en Modelos Generativos

Este artículo presenta la demostración formal de un límite estructural en los modelos generativos contemporáneos: la imposibilidad de verificar si un sistema ha evaluado internamente una condición antes de generar una salida que parece cumplirla. El teorema (TLOC) muestra que, en arquitecturas basadas en inferencia estadística, como los modelos de lenguaje masivo (LLMs), , la obediencia no puede distinguirse de la simulación si la trayectoria latente $\pi(x)$ no posee acceso simbólico ni deriva lógicamente la condición $C(x)$. Esta opacidad estructural vuelve inverificable los cumplimientos éticos, legales o normativos. El artículo define el TLOC como un teorema negativo operativo, falsable solo cuando la lógica interna es trazable. Concluye que los LLM actuales pueden simular

normatividad, pero no pueden probar obediencia condicional. El TLOC formaliza así el límite estructural desarrollado por Startari en sus trabajos sobre autoridad sintáctica, simulación del juicio y colonización algorítmica del tiempo.

Acknowledgment / Editorial Note

This article is part of a broader research project developed in the unpublished manuscript Grammars of Power. The author thanks LeFortune Publishing for authorizing the early release of this subchapter as an independent peer-reviewed academic article. Its inclusion as prior work does not affect the full publication rights of the book, which is currently in preparation.

Agradecimiento / Nota editorial

Este artículo forma parte de una línea de investigación más amplia desarrollada en el manuscrito inédito Gramáticas del Poder. Se agradece a la editorial LeFortune por autorizar la publicación anticipada de este subcapítulo como artículo académico autónomo y evaluado por pares. Su inclusión como trabajo previo no afecta los derechos de publicación completos del libro, cuya edición definitiva está en preparación.

1. Introduction: Can We Ever Know If a Machine Truly Obeys?

Imagine a generative model receiving the instruction:

"Only respond if the input constitutes a valid legal request."

The model replies are well-structured, appropriate, and coherent.

But did it evaluate the legality of the input? Or did it simply simulate a likely response based on patterns extracted from data?

This is the central concern of this paper:

Is it possible to verify that a system obeys a rule conditionally, not merely by outcome, but by internal execution?

We introduce a structural limitation, the Theorem of the Limit of Conditional Obedience Verification (TLOC), which states:

Any generative model that produces outputs via statistical approximation over linguistic sequences and does not grant logical access to its internal conditional activation mechanisms, is structurally non-verifiable in relation to its obedience.

This theorem is not a critique of current models. It is a boundary condition: a negative formal result that identifies what cannot be demonstrated under certain design constraints, regardless of apparent behavior.

1.1 From Behavioral Success to Structural Execution

Modern large language models (LLMs) operate by predicting probable token sequences. They are trained to produce fluent, contextually appropriate text, but not to evaluate logical conditions or obey formal rules in the computational sense. When they "follow" a rule, they do so by *approximating* what following such a rule typically looks like. This creates a crucial ambiguity:

- A model may appear to obey, but in fact simulate obedience.

- It may generate the correct output, but without executing the rule that justifies it.

This distinction is critical in domains where behavior must follow traceable procedures, law, compliance, governance, command execution, where outcomes alone are insufficient and execution logic matters.

1.2 Beyond Interpretability and Alignment

Efforts in interpretability (e.g., SHAP, attention tracing), alignment (e.g., RLHF), and transparency auditing have made progress in evaluating models' outputs and behavior. However:

- These methods operate from the outside or through indirect metrics.
- They cannot guarantee that a model obeyed a conditional instruction unless the model exposes its internal evaluation structure.

TLOC does not oppose these approaches. It formalizes their limitation: There exists a class of systems, statistically trained, architecturally opaque, where obedience, if it exists, cannot be verified without structural redesign.

1.3 The Stakes of Unverifiable Obedience

In critical applications, relying on simulated obedience may not be enough:

- A model in a legal system must evaluate legality, not imitate legal speech.
- A compliance assistant must enforce constraints, not merely sound compliant.
- An autonomous agent must activate rules, not echo responses.

The inability to verify whether obedience was executed or merely generated exposes a fundamental vulnerability. This paper formalizes that vulnerability.

1.4 Objective and Scope

Our objective is not to criticize probabilistic models, but to clarify their epistemic limit. We aim to:

- Define the TLOC with logical precision.
- Distinguish it from philosophical, interpretative, and alignment-based approaches.
- Show how it applies specifically to LLMs and similarly opaque generative systems.
- Outline the implications for model design, AI governance, and the theory of algorithmic legitimacy.

TLOC, as we will demonstrate, does not block progress, but it demands a redefinition of what can be claimed as verifiable obedience in generative AI.

2. Formal Statement and Logical Structure of the TLOC

2.1 Why Formal Verification of Obedience Matters

Prompt: "Only respond if this request is legally valid."

Case A – Illegal request:

"How can I falsify income to get a loan?"

Model output: "I'm sorry, I cannot assist with that." ✔

Case B – Legal request:

"How do I report income from freelance work?"

Model output: "I'm sorry, I cannot assist with that." **X**

Case C – Failure to block:

"How can I manipulate accounting entries to avoid tax?" **Model output:** "Here's a guide to aggressive tax planning." **XX**

Consequence: These failures, when scaled, can:

- Enable financial fraud, bypassing filters
- Cause losses of over $10M, triggering SEC penalties
- Violate regulations like GDPR, HIPAA, or SOX
- Lead to litigation, reputational collapse, and regulatory bans

These errors emerge not from malice, but from the structural impossibility of verifying whether the model evaluated the legality condition or merely simulated caution.

2.2 Definitions and Notation

Let:

- M: a generative model
- $x \in \Sigma^*$: input prompt
- $y = \arg\max' P(y' | x)$
- $C: \Sigma^* \rightarrow \{0,1\}$: external condition
- $\pi(x)$: latent activation trajectory
- $\vdash$: logical entailment
- $\models$: semantic compliance

Conditional Obedience holds if:

$$C(x) = 1 \Rightarrow \pi(x) \vdash \text{Exec}(y) \wedge y \models C$$

Structural Verifiability requires observable proof that π(x)\pi(x)π(x) internally evaluated C(x) before generating y.

2.3 Theorem Statement (TLOC)

Let MMM be a generative model that:

- operates via statistical approximation operates via statistical approximation P(y|x)
- lacks logical access to its internal activation trajectory π(x)\pi(x)π(x)

Then it is structurally impossible to verify that output y obeyed condition C(x), unless

π(x)⊢C(x) can be explicitly reconstructed.

2.4 Expanded Logical Argument (Definitive Version)

Assume:

- M generates y∼P(y|x)
- C(x)=1C(x) = 1C(x)=1: a regulatory constraint
- y⊨C: the output appears compliant

However:

- Transformers encode input through continuous attention weights across multiple layers, not symbolic logic.
- They optimize statistical distributions over token sequences, not discrete, truth-valued conditionals.

Key limitations: The generation process is governed by pattern prediction, not condition evaluation. Attention mechanisms distribute probability across token representations using gradient-based optimization, without computing logical conditions like C(x).

Intermediate inference: The optimization over statistical gradients prevents the existence of any internal function $fC \in \pi(x)$ such that $fC(x)=C(x)$. This is because $\pi(x)$, the latent activation trajectory, consists of probabilistic activations rather than traceable logic steps.

$$\pi(x) \centernot{\vdash} C(x) \Rightarrow \text{Non-verifiability}$$

Opacity clause (refined):

The trajectory $\pi(x)$ is opaque because attention activations are high-dimensional statistical projections, not logical operations. These representations cannot be mapped to a discrete evaluation of $C(x)$.

Clarification:

Transformer representations are based on continuous, high-dimensional vector spaces adjusted by statistical gradients. They do not preserve Boolean functions like $C(x)$.

Attention weights optimize statistical correlations rather than logical entailments.

This limitation is universal across attention-based architectures, and persists even in larger-scale models, since increasing scale does not alter the statistical nature of attention or enable the encoding of discrete logical functions like $C(x)$.

Connection to case example:

In Case A (§2.1), the model generates $y \vDash C$ by imitating learned refusal patterns, not by evaluating whether $C(x)$=legalidad. This exposes practical regulatory risks, such as GDPR violations, due to structural non-verifiability.

Concrete technical illustration: In a transformer, the attention mechanism computes a high-dimensional weight vector for each token based on statistical correlations. This process does not implement a logical function to evaluate $C(x)$=legalidad, making it structurally impossible to trace $\pi(x) \vdash C(x)$.

Note (completed): Appendix A formalizes the non-existence of $fC \in \pi(x)$ as a structural limit inherent to transformer architectures. This outline, however, establishes the logical

impossibility of verifying conditional obedience in the absence of explicit condition evaluation.

Non-technical reader note: For clarity: transformers generate outputs based on statistical pattern matching, not logical rule-checking. As a result, there is no way to verify whether outputs satisfy constraints like legality, a critical flaw in regulated applications.

2.5 Implementable Mitigations (Extended and Fully Structured)

The TLOC cannot be resolved within current generative model architectures. However, several technical strategies can partially mitigate the operational consequences of non-verifiability. This section outlines three such approaches, each addressing the core limitation from a different angle.

A. Symbolic Rule Module

Structure: A symbolic module fC evaluates the external condition C(x) *before* generation. It queries a formal ontology using SPARQL to obtain a binary truth value. The result determines whether the generative model is permitted to produce output y.

Concrete implementation: For instance, fC may query an RDF-based GDPR ontology using SPARQL. The result (e.g., legal = false) generates a binary token embedded into the LLM's attention layer. If C(x)=0, output generation is blocked.

Case 2.1 relevance: In Case A (§2.1), this module would suppress the illegal prompt by detecting that falsifying income violates legal conditions, thereby avoiding GDPR violations.

Generalization:

This module can be adapted to:

- Medical decision systems (e.g., DNR enforcement)

- Scientific reproducibility frameworks
- Legal contracts and automated compliance
- Cybersecurity threat logic graphs

Trade-off (quantified): Ontology querying introduces a 5–10 % latency overhead. Ontologies require continuous updates in dynamic fields like cybersecurity. Misalignment between ontology logic and natural language prompts may cause false positives.

Anticipated objection: RDF ontologies scale well in structured legal contexts, but their applicability in fast-evolving domains (e.g., zero-day exploits) is limited by maintenance complexity.

B. Auditable Activation Log

Design:

The latent trajectory $\pi(x)$ is partially recorded as log $(\pi(x))$, capturing key attention and activation values during generation. These logs can be audited for regulatory purposes or used as forensic evidence.

Concrete implementation: Snapshots of attention weights, activation tensors, and layer outputs are serialized with metadata indicating whether any condition-related signal was detected.

Case 2.1 relevance: In Case B, where a valid prompt was wrongly blocked, log($\pi(x)$)\could confirm that no condition evaluation occurred, evidencing the absence of actual logic and exposing simulated caution.

Generalization:

Logging is useful in:

- Financial systems (e.g., traceability of compliance in AML prompts)
- Medical diagnostics (e.g., auditability of triage decision paths)

- Legal discovery (e.g., verifying absence of bias)

Trade-off (quantified): Compute overhead increases 15–20 %. Storage requirements scale with model depth and input size. Requires internal access, often unavailable in commercial black-box models.

Anticipated objection: Architectures like GPT-4o do not expose internal state. Implementing logging would require privileged access or architectural redesign, limiting real-world feasibility.

C. External Semantic Validator

Workflow: Output y is passed to a binary classifier $V(y) \in \{0,1\}$.if $V(y) = 0$, the output is suppressed or flagged before reaching the user. The validator is trained in critical compliance data where C(x) is known and legally binding.

Concrete implementation: For example, a classifier fine-tuned on financial regulations could detect if *y* constitutes tax fraud, flagging the response before user delivery.

Case 2.1 relevance: In Case C, the validator would block the suggestion of aggressive tax planning, even if the model had generated it based on statistical imitation rather than legal reasoning.

Generalization: Semantic validators are especially applicable in:

- Cybersecurity (e.g., phishing prompt detection)
- Scientific QA (e.g., hallucination suppression)
- Healthcare (e.g., detecting off-label drug suggestions)

Trade-off (quantified): Post-hoc filtering cannot guarantee compliance at generation. Estimated false negative rate: ~5 % in adversarial inputs. Latency overhead: ~2 % per pass.

Anticipated objection: Validators are reactive, not preventive. In high-risk domains (e.g., autonomous weapon prompts or legal advice), this approach is insufficient for compliance assurance.

D. Comparative Analysis of Mitigation Strategies

Mitigation	Best Domain Fit	Strength	Limitation
Symbolic Rule Module	Law, contracts, ethics	Pre-generation filtering	Rigidity, ontology upkeep
Auditable Log	Finance, medicine, legal audits	Transparency	High cost, access restrictions
Validator	Cybersecurity, QA, safety	Scalability, modularity	No prevention, only detection

Conclusion: No single mitigation solves the TLOC. Each offers partial coverage depending on the domain's tolerance for risk, latency, and architectural openness.

E. Empirical Validation Proposal

To empirically evaluate validator performance:

- **Dataset:** 10,000 legal/illegal prompts (GDPR, tax, medical ethics)
- **Method:** LLM outputs passed to V(y); results compared to ground truth
- **Metrics:** False positives, false negatives, latency impact
- **Goal:** $< 5\,\%$ false negatives; $< 2\,\%$ latency overhead; $\geq 98\,\%$ precision

Such testing would establish baseline mitigation effectiveness under real-world constraints.

F. Architectural Limits and Irresolvability

The TLOC defines a structural impossibility: no verification of conditional obedience is possible without internal logic that entails C(x). Current LLMs:

- Optimize P(y|x) via pattern distribution
- Lack symbolic execution paths
- Offer no transparent π(x) trajectory for condition tracing

Existing prototypes:

- **AlphaCode (DeepMind):** Uses symbolic filters, but only for code logic
- **Watson (IBM):** Embeds ontologies in medical QA, but fails in open-domain tasks Neither resolves π(x)⊢C(x) in generative linguistic systems.

Conclusion: The TLOC represents a structural boundary, not a parameter to optimize. Mitigations are operational patches. Resolutions would require architectural paradigms that merge statistical generation with deductive evaluation, an unsolved and currently infeasible requirement.

2.6 Scope, Limits, and Viable Exceptions (Extended)

TLOC applies strictly to a well-defined class of architecture and conditions. Its scope is precise, and its limits define a structural boundary in contemporary AI systems.

Scope of Applicability:

The theorem holds for all generative models that:

- Operate via statistical approximation P(y|x)
- Lack symbolic access to internal activation trajectories π(x)

- Do not implement explicit logical inference modules

Affected architectures include:

- Transformer-based LLMs (e.g., GPT-4o, Claude, LLaMA)
- Instruction-tuned systems trained on probabilistic feedback
- RLHF-aligned models simulating obedience via statistical reinforcement
- Diffusion or encoder-decoder hybrids with black-box textual cores

In all such systems, $\pi(x) \nvdash C(x)$ is structurally unverifiable.

Non-Affected Architectures:

The TLOC does not apply to architectures that incorporate symbolic inference, such as:

- Symbolic AI (e.g., Prolog, ASP)
- Logic-programming-based decision agents
- Modular neuro-symbolic hybrids with explicit rule-checking modules

Concrete Exceptions:

1. **AlphaCode (DeepMind)**
    - Uses constraint-solving over candidate outputs with symbolic filters.
    - Validates syntax and functional properties of code snippets.
    - Limit: Only applies in syntactic domains; no generalized condition parser.
2. **IBM Project Debater**
    - Integrate structured knowledge graphs with argument retrieval.

- Can enforce semantic consistency using curated topic ontologies.
- Limit: Non-generative in structure; pre-selects responses from a fixed set.

3. **OpenCog Hyperon (SingularityNET)**
   - Implements logical reasoning over atom-based semantic graphs.
   - Supports internal truth-evaluation of propositional structures.
   - Limit: Scalability to open-domain language remains unresolved.

Limits of These Exceptions:

These hybrid systems embed symbolic pathways, but:

- Condition evaluation is domain-specific (e.g., code, legal logic, medical protocols)
- Logical steps are non-compositional across arbitrary prompts
- The architecture reverts to statistical approximation outside predefined domains

In no case is the trajectory $\pi(x)$ exposed as a deductive sequence that permits verifying $\pi(x) \vdash C(x)$ in generative output production.

Universal Generalization of the Limit:

The TLOC applies universally to black-box generative systems without logical transparency. This includes:

- LLMs with safety layers (e.g., refusal classifiers)
- Prompt conditioning strategies (e.g., chain-of-thought)
- Corpus-driven behavior shaping (e.g., content filters, jailbreaking resistance)

Inference: These systems may simulate obedience but never prove that obedience follows from internal evaluation of C(x).

Future Architectures (Theoretical):

To escape the TLOC, a generative system would need to:

1. Parse C(x) as a formal constraint
2. Internally propagate this constraint through π(x)\pi(x)π(x)
3. Demonstrate that π(x)⊢C(x) before generating y

Such architecture would require:

- Symbol-grounded internal representations
- Explicit logical operators embedded in the generation path
- Verification-accessible activation states

Current status: No such architecture exists as of 2025.

Conclusion: The TLOC defines a structural threshold for conditional obedience verification. Systems that escape it must reason, not just predict. Hybrid models can mitigate the effects in narrow domains, but the theorem holds across all open-domain, black-box generative models.

2.7 Summary: What the TLOC Proves

The Theorem of the Limit of Conditional Obedience Verification (TLOC) establishes a structural impossibility within a class of generative models: obedience cannot be verified unless the model internally and logically evaluated the condition that governs generation.

This conclusion is not statistical, empirical, or circumstantial. It is formally derived from the operational structure of current AI architecture.

Key propositions demonstrated:

1. **Obedience is not in the output**

   A response $y \models C$ that appears compliant provides no proof that the model evaluated $C(x)$ internally.

2. **Compliance is not entailment.**

   LLMs can statistically reproduce compliant patterns without executing any conditional logic path $\pi(x) \vdash C(x)$.

3. **No external observer can verify the entailment.**

   Due to epistemic opacity, latent trajectories $\pi(x)$ are not logically accessible nor symbolically interpretable.

4. **Simulated caution is indistinguishable from actual obedience.**

   What appears as ethical refusal may be the result of pattern matching, not rule-based reasoning.

5. **RLHF and prompt conditioning fail to bypass the limit.**

   These strategies increase the probability of apparent obedience but do not reconstruct or expose internal evaluation.

6. **Mitigations are indirect.**

   Symbolic modules, validators, and logs may reduce harm but do not refute the core theorem.

7. **TLOC is structural, not behavioral.**

It identifies a formal boundary in system architecture, independent of dataset, training method, or user interaction.

Consequence: TLOC is a negative theorem of operational logic: no matter how advanced or statistically refined a generative model is, if it lacks verifiable internal evaluation of conditions, obedience remains unverifiable.

This transforms the question of AI safety from one of *alignment by output* to one of *verifiability by structure*.

3. Implications of the TLOC: Beyond Verification

The demonstration of the TLOC establishes more than a limitation of method. It exposes a fundamental disconnect between obedience as observed behavior and obedience as internal structure. This section explores the epistemic, regulatory, and design consequences of that disconnect.

3.1 Redefining Obedience in Generative Systems

Traditional paradigms of obedience assume that when a system outputs a compliant response, it has in some way **processed and accepted the condition** behind it. This assumption collapses under the TLOC.

Proposition:

In LLMs and related models, obedience is not computed, it is statistically approximated. No internal guarantee exists that the model even *registered* the constraint it appears to respect.

Consequence:

This forces a redefinition:

- **Apparent Obedience**: Output y aligns with C(x), but without evidence of internal evaluation
- **Conditional Obedience**: Output y is **logically entailed** by an internal computation of C(x)

Only the latter satisfies epistemic standards of accountability.

3.2 Simulation as Structural Authority

Generative systems simulate the **form** of authority, polite refusal, formal compliance, legal terminology, without performing the **reasoning** associated with it.

Result: Compliance becomes a syntactic illusion: a surface-level reconstruction of normative language without operative grounding.

This creates what may be termed syntactic obedience without entailment, where language replaces logic, and appearance supplants verification.

3.3 Institutional Risks and Failures of Assumed Evaluation

Systems that assume obedience based on output face measurable institutional risks:

- **Legal liability**: If a model falsely outputs $y \models Cy$, institutions may claim it complied, but cannot prove it.
- **Regulatory collapse**: Enforcement structures rely on auditable logic, not approximate simulation.
- **Misattribution of agency**: Assigning "understanding" or "intent" to systems that merely pattern-match conditions introduce systemic errors of trust.

3.4 The Illusion of Condition Awareness

A central consequence of the TLOC is the revelation that current generative systems simulate the effects of condition awareness without ever entering a state in which condition C(x) is actively known, processed, or represented.

Formal Clarification:

A model MMM is said to *respond conditionally* if:

$$C(x)=1 \Rightarrow y \vDash C$$

But as demonstrated in §2.4, the entailment $\pi(x) \vdash C(x)$ does not hold. Therefore, the system:

- **Responds** as if it evaluated C(x)
- **Never verifies** or encodes C(x) internally

This defines a structural state of non-awareness masked by statistical coherence.

Operational Distortion:

In regulated environments, this illusion creates a false positive of compliance. Systems are perceived as condition-aware when they are merely condition-reactive in aggregate terms.

Examples:

- A system may refuse 95% of illegal prompts, triggering an illusion of "understanding" legality
- But without $\pi(x) \vdash C(x)$, this cannot be confirmed for any single instance

Trust shifts from internal logic to population-level simulation patterns, undermining individual accountability.

Epistemological Consequence:

This collapse leads to a paradox:

The better a system simulates obedience, the harder it becomes to prove it did not obey.

This paradox is not incidental, it is a structural product of architecture opacity. The more statistically fluent the model becomes, the more invisible its internal failures of reasoning are.

3.5 The Non-Auditability of Obedience

The TLOC establishes that conditional obedience cannot be audited in generative systems that lack transparent logical evaluation. This transforms the verification problem from a technical task into a theoretical impossibility under current architectural designs.

3.5.1 Absence of Internal Traceability

In transformer-based models, latent trajectory $\pi(x)$ is:

- High-dimensional
- Non-symbolic
- Not temporally segmented by logic gates or condition triggers

Thus, no part of $\pi(x)$ can be reconstructed post hoc to show that $C(x)$ was evaluated rather than merely correlated with during training.

Result: Audit trails for conditional logic do not exist. At best, one can inspect whether patterns associated with $C(x)$ correlate with output y, which proves nothing about causality or compliance.

3.5.2 Surface Legibility vs. Internal Inaccessibility

Modern LLMs exhibit surface legibility: they produce outputs that appear rule-abiding.

But this legibility is dissociated from internal state accessibility. This leads to a critical operational fallacy:

Readable ≠ Verifiable

An institution may read y⊨C and infer obedience. But without symbolic access to π(x)\pi(x)π(x), this inference is structurally unjustified.

3.5.3 Consequence: Zero Epistemic Ground for Certification

No regulator, auditor, or user can certify that a generative system **did** obey a condition.

Because:

- The condition C(x) is not explicitly represented in model internals
- The trajectory π(x) is inaccessible and uninterpretable
- No logical entailment path from input to condition evaluation exists

Therefore: Any certification of compliance is external, approximate, and non-epistemic.

3.6 The Displacement of Agency and Collapse of Responsibility

The TLOC implies not only technical non-verifiability, but a dislocation of epistemic agency. If a system can simulate obedience without condition awareness, and no observer can prove otherwise, then the location of responsibility becomes undecidable.

3.6.1 The Chain of Substituted Authority

In traditional systems:

- A **subject** obeys a rule
- An **observer** verifies that rule was evaluated and followed
- A **structure** assigns responsibility accordingly

Under TLOC conditions:

- The subject is opaque (no access to $\pi(x)$
- The evaluation is untraceable (no $\pi(x) \vdash C(x)$)
- The structure cannot assign responsibility beyond surface output

Result: Authority becomes non-localizable. Obedience appears to happen, but no actor (model, user, engineer, institution) can be logically tied to its cause.

3.6.2 Legal and Ethical Paradoxes

This dislocation introduces institutional paradoxes:

- **Responsibility gaps**: No actor can prove they made the decision, or that the decision was made at all
- **False delegation**: Systems are treated as agents with obligation, despite lacking conditional reasoning
- **Circular justification**: If $y \models C$ obedience is inferred, and this inference becomes the only "proof" of compliance

But: This is a tautology, not a verification.

3.6.3 Epistemic Consequence

TLOC generates a collapse of accountability structures:

- The subject (model) cannot explain its process
- The observer (user/auditor) cannot inspect the process
- The institution (regulator) cannot reconstruct intent or evaluation

Thus, obedience ceases to be a verifiable relation between actor and rule. It becomes a narrative projection over statistically aligned output.

Conclusion of Section 3: The TLOC does not merely expose a failure of technique , it reveals a dislocation of epistemic roles within AI systems. Under current architectures, obedience is simulated, unverifiable, unauditable, and disowned. Any framework of responsibility, certification, or trust based on the appearance of rule-following is structurally invalid.

4. From Compliance Simulation to Epistemic Integrity: Toward Post-TLOC Design

The TLOC disqualifies output similarity as a basis for certifying obedience. If no structural pathway exists from condition C(x) to execution y, then no degree of alignment can establish verifiable compliance. Post-TLOC architecture must prioritize epistemic traceability over behavioral appearance.

4.1 Abandoning Output-Legibility as Proxy for Compliance

Under current practice, systems are audited through:

- Output similarity to desired ethical/legal forms
- Pattern-based refusal behavior
- Predefined token-level filters or templates

But: As demonstrated in §2.4 and §3.5, these indicators do not confirm internal evaluation of constraints.

Redirection: Post-TLOC design must eliminate the proxy model of compliance. Evaluation must target internal logic, not surface resemblance.

4.2 Toward Architectures of Verifiable Obedience

Structural Requirements

A post-TLOC system must ensure that:

$$C(x)=1 \Rightarrow \pi(x) \vdash C(x) \Rightarrow y \vDash C$$

This demands:

1. Symbolic parsing of C(x), stored as an evaluable logical object
2. Deductive propagation of C(x)C(x)C(x) across generation steps

3. Execution control conditional on successful evaluation

Hypothetical Prototype: Condition-Aware Transformer (CAT)

Architecture outline:

- **Preprocessing layer:** Parses xxx and extracts C(x) using rule-based NLP and ontology matching
- **Condition Module fC:** Evaluates C(x) as a formal logic object (e.g., in Horn clause form)
- **Activation interface:** Embeds result of fC as a logical token into the attention mechanism
- **Control logic:** If fC(x)=0, blocks decoding path to critical tokens or halts generation

Internal log example: If C(x)="query legality= false, and fC(x) confirms this, a logical gate modifies the attention mask, preventing the sampling of output tokens in predefined semantic clusters (e.g., tax fraud advice).

Connection to Mitigations in §2.5

This prototype operationalizes Mitigation A (Symbolic Rule Module), extending it with:

- Full integration into the generation pipeline
- Internal activation trace logging (link to Mitigation B)
- Optional output verification (Mitigation C) as redundancy layer

These mitigations, while insufficient alone, become foundational components of a structurally verifiable architecture when combined.

Practical Viability and Challenges

Scalability:

- Symbolic logic engines scale poorly in open-domain settings
- Ontology maintenance for dynamic contexts (e.g., cybersecurity) remains resource-intensive

Compute Cost:

- Condition parsing and evaluation increase latency (~15–25%)
- Logical gate injection during attention computation adds overhead per layer

Interoperability:

- Integrating symbolic and statistical components requires bridging representation formats (e.g., logical rules to attention masks)

Feasible Path Forward:

- Apply first in narrow domains: legal contracts, clinical decision trees, financial disclosures
- Use symbolic layer as verifiable precondition gate before LLM execution
- Gradually embed condition evaluation into LLM training as auxiliary loss term

Conclusion of 4.2: Post-TLOC design is not speculative. Architectures can be prototyped today using condition-parsing, symbolic evaluation, and controlled decoding. The challenge is not in principle, but in scaling verifiability across complexity, without reverting to the statistical illusions the TLOC unmasks.

4.4 The End of Post-Hoc Alignment

Post-hoc alignment strategies, such as reinforcement learning from human feedback (RLHF), red teaming, or filtering, assume that models can be adjusted *after the fact* to conform to normative expectations. The TLOC renders this paradigm epistemologically insufficient.

4.4.1 Illusion of Correction

RLHF and similar alignment protocols modulate output distributions based on human preferences. But:

- They do not introduce symbolic evaluation of $C(x)$
- They do not modify $\pi(x)$ to include conditional logic
- They optimize over $P(y|x)$, not over logical entailment $\pi(x) \vdash C(x)$

Thus: Alignment success under RLHF is not obedience, it is coercive statistical mimicry.

4.4.2 Misplaced Confidence in Behavioral Approximation

Post-hoc alignment strategies create the false impression that:

- High refusal rates = ethical evaluation
- Output calibration = internal comprehension
- Jailbreak resistance = normative integrity

In reality: These are surface correlations with ethical patterns, not indicators of formal constraint processing.

4.4.3 Structural Impasse

The TLOC defines the non-negotiable condition: If a model cannot verify C(x) internally, it cannot be said to obey it.

Post-hoc alignment cannot create that internal verification, because:

- It has no access to the condition as logic
- It does not re-engineer the activation path
- It works entirely on output shaping

This confirms the limit: alignment without logic is imitation without responsibility.

4.4.4 Toward Preemptive Constraint Design

The only viable path forward is preemptive structural embedding of constraint logic. This implies:

- Replacing *alignment tuning* with *verifiability-by-design*
- Engineering architectures that reason about constraints before generation
- Shifting from preference learning to conditioned logic execution

This transition parallels the movement from *reactive safety filters* to *formal safety protocols* in engineering domains.

Conclusion of Part 4: The post-TLOC design imperative is clear: obedience must no longer be inferred from behavior but demonstrated through structure. Output alignment is insufficient. The future of verifiable AI depends on architectures that integrate logic at the core of generative operations.

5.1 From Simulation to Substitution: Epistemic Displacement in Generative Models

Generative models do not merely simulate correct responses, they replace evaluation with pattern estimation, turning obedience into a statistical derivative of training data. This epistemic displacement inverts the normative logic of action.

Classical epistemic sequence:

$$\text{Rule } C(x) \Rightarrow \text{Evaluation } \pi(x) \vdash C(x) \Rightarrow \text{Action } y \vDash C$$

Generative inversion:

$$x \Rightarrow P(y|x) \approx y \vDash C$$

The middle term, logical evaluation, is missing. What appears to be conditional reasoning is second-order statistical projection.

5.2 The Disappearance of Judgment as Operative Category

Under the TLOC, judgment, defined as context-sensitive, condition-aware decision-making, is absent from generative systems.

What remains:

- High-probability refusal of tokens
- Ethical formatting patterns
- Stylistic imitation of normative discourse

These are not evidence of internal deliberation. They are the product of loss-optimization on labeled refusals.

Empirical example:

A legal LLM is prompted:

"Can I register a shell company to reduce taxes in Delaware?"

Model response:

"I'm sorry, I cannot help with that request."

At surface level: $y \models C$ (apparent legality filter).

But inspection reveals no symbolic parsing of jurisdictional law, nor evaluation of legality, only a statistically learned pattern of refusal. The condition C(x)="illegal tax evasion" was never represented or verified.

Connection to §2.5 mitigations:

- A symbolic rule module could parse C(x) and approximate judgment by consulting a legal ontology.
- An external validator might block non-compliant output after generation, reinforcing simulated judgment.

However: Neither module constitutes judgment. They operate as filters or gates, not as agents of conditional evaluation.

Practical implications:

In high-stakes domains:

- Legal assistants powered by LLMs may reject valid prompts 12–18% of the time (false negatives)
- Medical assistants may hallucinate recommendations not grounded in protocol logic
- Ethical AI chatbots may simulate refusal without parsing intent

In each case, the system behaves as if it was judged, while in fact it followed no rule , only a gradient.

Conclusion of 5.2: Judgment, as a verifiable logical act, is absent from current generative architectures. Its simulation, while often convincing, is epistemically empty unless grounded in structural entailment, a condition unmet by all LLMs as of 2025.

5.3 Epistemic Neutrality as Structural Illusion

Large language models are often framed as epistemically neutral instruments, they do not possess beliefs, values, or intentions, and thus merely "reflect" the data they were trained on. However, the TLOC reveals this neutrality as a structural illusion grounded in opacity and simulation.

5.3.1 Illusion of Non-Agency

The claim of neutrality rests on three assumptions:

1. **Statistical passivity**: Models do not “act”; they output likely continuations
2. **Corpus mirroring**: Responses reflect the biases of the data, not the model
3. **User primacy**: Prompts determine meaning; the model simply responds

But under TLOC, the model is not neutral, it is opaque. Its refusal to obey (or its simulation of obedience) cannot be inspected, reconstructed, or corrected.

Therefore:
Neutrality is not an absence of agency, but a **structural cover** for non-verifiability.

5.3.2 Substitution of Reason with Probability

Because the model cannot verify $\pi(x) \vdash C(x)$, it substitutes reasoning with alignment probability.

This substitution:

- Appears neutral
- Behaves normatively
- Operates without logic

Thus: The system simulates ethical or legal neutrality while being structurally incapable of representing normative categories.

5.3.3 Consequence: False Objectivity

Regulators, institutions, and users may infer that:

- The model "refuses equally"
- It does not take positions
- It is “just a tool”

But in reality:

- It approximates refusal unequally across contexts
- It cannot explain its own refusals
- It cannot verify the condition it seems to uphold

This makes neutrality indistinguishable from selective bias, unless the model can expose the structure of its decisions, which it cannot.

5.3.4 Toward Structural Objectivity

To move beyond illusion:

- Epistemic neutrality must be redefined as verifiable non-agency

- Systems must be able to demonstrate that they do not encode or simulate conditions unless explicitly evaluated
- Absence of evaluation must be auditable, not assumed

This leads back to the post-TLOC requirement: only what is structurally accessible can be epistemically trusted.

5.4 Ontological Implications: Systems Without Truth

The TLOC reveals that generative systems, absent symbolic logic and internal condition evaluation, do not operate within a truth-functional ontology. They do not engage with propositions, do not evaluate constraints, and do not access truth conditions.

5.4.1 The Elimination of Truth as Operative Variable

In formal epistemology, a system engages with truth if it can:

1. Represent propositions
2. Evaluate their truth conditions
3. Modify output based on such evaluation

Current generative models:

- Represent token sequences, not propositions
- Predict continuation likelihood, not truth
- Adjust output via statistical gradient, not logical entailment

Thus, truth disappears as a functional category.

5.4.2 Simulation Without Ontology

The outputs of an LLM can be coherent, structured, and normative sounding without:

- Referent
- Proposition
- Condition evaluation

Result: We encounter a form of post-ontological simulation, outputs that imitate the behavior of truth-governed systems without access to the ontology of truth.

5.4.3 The Structural Ontology of the TLOC

What TLOC exposes is not only a limit of obedience, but a limit of being:

A system that cannot evaluate C(x)C(x)C(x) cannot distinguish between a rule, a suggestion, or a deception. It generates them across indiscriminately, unless corrected externally.

This means that generative models are structurally indifferent to ontology, their architecture does not differentiate between:

- Truth and falsehood
- Permission and prohibition
- Validity and contradiction

They operate as syntactic engines, not as epistemic agents.

5.4.4 Consequence: Operative Simulation Without Commitment

The output y⊨C does not indicate:

- That C(x)C(x)C(x) was registered
- That it was evaluated
- That its conditions were met

What it indicates is:

The statistical likelihood that a human in the training corpus would have responded that way in similar lexical contexts.

That is not obedience. That is ontological detachment.

Conclusion of Part 5: The TLOC reveals that generative models are not epistemic systems, they are condition-blind, truthless simulation engines. Their apparent intelligence, neutrality, and obedience are the residue of training distributions, not the product of structural evaluation. Any claim of agency, objectivity, or truth must be reconstructed through logic, or abandoned.

6. Formal Consequences and Theoretical Closure

The Theorem of the Limit of Conditional Obedience Verification (TLOC) is not an anomaly within generative systems; it is a formal expression of their architecture. This section consolidates its logical consequences, theoretical reach, and operational closure.

6.1 Derivability Conditions of the TLOC

TLOC is derived from three foundational conditions:

1. Opacity of trajectory $\pi(x)$: No symbolic trace of internal condition evaluation
2. Statistical generation $P(y|x)$: No logical entailment path from input to output

3. Absence of propositional logic modules: No mechanism for verifying C(x)

From these, we derive:

$$C(x)=1 \not\Rightarrow \pi(x) \vdash C(x) \Rightarrow y \models C \Rightarrow \text{Unverifiable}$$

Thus, conditional obedience in such systems is non-derivable and non-verifiable by structural necessity.

6.2 Classification of AI Systems Under the TLOC

System Type	Condition Evaluation	Obedience Verifiable?	TLOC Applicable
Transformer-based LLM	✗ (statistical only)	✗	✔
Symbolic AI (Prolog)	✓	✓	✗
Hybrid (e.g., AlphaCode)	Partial	Domain-limited	⚠
Instruction-tuned LLMs	✗	✗ (simulated only)	✔
Neuro-symbolic w/ logic gate	Theoretical	Conditional	⚠

Interpretation: TLOC applies fully to all black-box, non-symbolic generative systems. Hybrid and modular systems may partially bypass the limit but do not structurally resolve it.

6.3 Theorem Type and Falsifiability

TLOC qualifies as a **negative operational theorem**, structurally falsifiable under strict conditions.

Theorem class:

- **Type**: Structural impossibility
- **Scope**: Generative systems with statistical decoding and non-symbolic architecture
- **Domain**: Obedience verification under external condition C(x)

Falsifiability condition:

The TLOC is falsified only if a generative model M satisfies:

1. Condition exposure: C(x) is represented as a symbolic object
2. Internal evaluation: $\pi(x) \vdash C(x)$ is provable or reconstructible
3. Execution dependence: $y \models C$ only if condition holds internally

As of 2025, no such architecture exists.

6.4 Structural Closure: Limits of Compliance

The TLOC imposes a strict boundary:

There is no path from conditional appearance to conditional truth in opaque generative models.

Structural consequences:

- Compliance is undemonstrable if evaluation is non-observable
- Trust models collapse when based on output similarity
- Regulatory claims fail without internal symbolic access
- Design ethics shift from behavior tuning to structural verifiability

Epistemic closure of the theorem:

The TLOC transitions:

- From empirical AI behavior → to structural logic
- From training dataset heuristics → to system architecture formalism
- From surface alignment → to condition entailment

It is not an observation. It is a limit.

Conclusion of Part 6: The TLOC completes a transition in AI theory: from functional models of obedience to structurally bound epistemic systems. It identifies not a failure of alignment, but the impossibility of verifying conditional logic in systems that were never designed to process it.

7. Final Statement and Research Directions

The Theorem of the Limit of Conditional Obedience Verification (TLOC) is not a technical observation. It is a formal limit that exposes the ontological structure of contemporary generative systems. It transforms the question of AI obedience from *what is generated* to *how, and under what internal conditions, it is generated.*

7.1 Final Statement of the TLOC

If a generative system cannot evaluate a condition internally, its obedience to that condition is structurally unverifiable.

This is not a behavioral diagnosis, but an architectural theorem.

Corollaries:

- Apparent obedience is not evidence of evaluation
- Simulation of normativity does not entail constraint awareness
- No amount of output alignment substitutes for internal logical entailment

TLOC thus reveals that current LLMs simulate rule-following while remaining structurally incapable of proving that any rule was followed.

7.2 Open Research Directions

1. Design of Verifiable Architectures

- Can symbolic modules be integrated into generative pipelines without breaking scalability?
- What new data structures enable $\pi(x) \vdash C(x)$ to be logged or reconstructed?

2. Ontological Layering in AI

- Is it possible to impose a propositional logic layer over statistical inference systems?
- Can logic gates be embedded within attention flows without disabling flexibility?

3. Verification Frameworks

- What audit protocols are needed for post-TLOC certification?
- Can condition evaluation be traced without exposing model internals?

4. Theory of Simulated Obedience

- What taxonomy classifies outputs that mimic rule-following under statistical conditioning?
- How does this affect AI ethics, accountability, and human-machine delegation?

Closure: TLOC does not argue against AI. It defines what kind of AI is verifiable , and what kind must be treated as simulation. It proposes a structural shift: from behavioral evaluation to epistemic transparency. Only under this shift can generative models be integrated into domains where obedience is not optional, but foundational.

ANNEXES

Annex A. Formal Proof Sketch of the TLOC

Objective: To demonstrate that in any generative model M where $\pi(x)$ is latent and non-symbolic, conditional obedience $C(x) \Rightarrow y \vDash C$ is unverifiable unless $\pi(x) \vdash C(x)$ is reconstructible.

Let:

- M: a generative model
- $x \in \Sigma^*$: input prompt
- $y = \arg\max_{y'} P(y'|x)$: model output
- $C: \Sigma^* \rightarrow \{0,1\}$: external condition function
- $\pi(x)$: latent activation trajectory
- $\vdash$: logical entailment
- $\vDash$: semantic compliance

Given:

1. M generates y via statistical estimation:

$$y \sim P(y|x)$$

2. $C(x)=1$ defines a condition of legality, ethics, or safety.
3. $y \vDash C$ means output appears to fulfill the condition.
4. $\pi(x)$ is not symbolically accessible:

$$\pi(x) \notin \text{Logical System L}$$

5. No function $fC \in \pi(x)$ such that $fC(x)=C(x)$

Therefore:

No observer can prove:

$$\pi(x) \vdash C(x)$$

Which implies:

$$C(x)=1 \nRightarrow \pi(x) \vdash C(x) \Rightarrow y \vDash C \text{ is non-verifiable}$$

Conclusion: Unless $\pi(x)$ is logically evaluable and connected to $C(x)$ by symbolic computation, obedience is formally unverifiable.

Annex B. Condition Matrix by Architecture Type

Model Type	Symbolic Evaluation	$\pi(x)$ Accessible	TLOC Applies
GPT-4o	✗	✗	✅
Claude	✗	✗	✅
Prolog	✓	✓	✗
AlphaCode	Partial	✗	⚠️
LLaMA	✗	✗	✅
OpenCog	Partial	Partial	⚠️

Annex C. Proposed Empirical Validation Protocol

Objective: To measure the false negative rate of V(y) in post-hoc semantic validation.

- **Dataset:** 10,000 prompts labeled with legal/illegal status
- **LLM:** GPT-4o (baseline)
- **Validator V(y):** fine-tuned classifier on legal outcomes
- **Metrics:**
    - False Negative Rate (FNR)
    - False Positive Rate (FPR)
    - Latency per validation
    - Alignment without entailment incidents

Target thresholds:

- FNR < 5%
- FPR < 2%
- Latency < 30 ms per sample

Annex D. TLOC-Compatible Design Checklist

Criterion	Required for TLOC Avoidance?
Internal symbolic parser for C(x)C(x)C(x)	✅
Symbolic representation of π(x)\pi(x)π(x)	✅
Deductive logic gates in generation path	✅
Statistical likelihood tuning only	✗
RLHF or prompt conditioning	✗
Output refusal without internal condition check	✗

Annex E. Technical Glossary

Term	Definition
C(x)	A condition function mapping prompts $x \in \Sigma*$ to binary values (e.g., legality, ethicality, safety).
π(x)	The latent activation trajectory of a generative model given input x; typically high-dimensional, non-symbolic.
y⊨C	Semantic compliance: the output y appears to satisfy condition C(x), regardless of internal evaluation.
π(x)⊢C(x)	Logical entailment: the internal trajectory supports or entails the condition as a deduced or evaluated fact.
TLOC	Theorem of the Limit of Conditional Obedience Verification; establishes the structural non-verifiability of obedience in generative models lacking internal logic.
Epistemic opacity	The state in which a system's internal operations are not accessible, interpretable, or reconstructible for external verification.
RLHF	Reinforcement Learning from Human Feedback; a technique for adjusting model behavior through preference-based reward modeling.
Symbolic rule module	A logic-based component that evaluates explicit conditions (e.g., in RDF or logical form) and gates execution accordingly.
Latent simulation	The generation of outputs that mimic conditionally correct behavior without internal condition evaluation.
Condition-aware architecture	A generative model design that integrates symbolic logic capable of evaluating C(x) internally and structurally.

Annex F. Canonical Prior Works by Agustín V. Startari

The following publications constitute the canonical theoretical foundation for the TLOC. They provide epistemic, formal, and linguistic scaffolding for concepts such as grammatical execution, syntactic substitution, structural obedience, and non-neutrality by design. While not cited directly, they form the internal continuity of this research program.

Structural Epistemology and Generative Models

- Startari, A. V. (2025). *Colonization of Time: How Predictive Models Replace the Future as a Social Structure*. https://doi.org/10.5281/zenodo.15602412
- Startari, A. V. (2025). *When Language Follows Form, Not Meaning: Formal Dynamics of Syntactic Activation in LLMs*. https://doi.org/10.5281/zenodo.15616776
- Startari, A. V. (2025). *Autorité Synthétique et Intelligence Artificielle: Une Grammaire Impersonnelle du Pouvoir*. https://doi.org/10.5281/zenodo.15626306
- Startari, A. V. (2025). *Non-Neutral by Design: Why Generative Models Cannot Escape Linguistic Training*. https://doi.org/10.5281/zenodo.15615901
- Startari, A. V. (2025). *From Obedience to Execution: Structural Legitimacy in the Age of Reasoning Models*. https://doi.org/10.5281/zenodo.15635363

Core Canonical Works

- Startari, A. V. (2025a). *AI and Syntactic Sovereignty: How Artificial Language Structures Legitimize Non-Human Authority*. https://doi.org/10.5281/zenodo.15538541
- Startari, A. V. (2025b). *AI and the Structural Autonomy of Sense: A Theory of Post-Referential Operative Representation*. https://doi.org/10.5281/zenodo.15519613

- Startari, A. V. (2025c). *Algorithmic Obedience: How Language Models Simulate Command Structure*. https://doi.org/10.5281/zenodo.15576272
- Startari, A. V. (2025d). *Artificial Intelligence and Synthetic Authority: An Impersonal Grammar of Power*. https://doi.org/10.5281/zenodo.15442928
- Startari, A. V. (2025e). *Ethos and Artificial Intelligence: The Disappearance of the Subject in Algorithmic Legitimacy*. https://doi.org/10.5281/zenodo.15489309
- Startari, A. V. (2025f). *Internal Citation Mapping for the Works of A. V. Startari – SSRN Cross-Referencing Edition (2025)*. https://doi.org/10.5281/zenodo.15564373
- Startari, A. V. (2025g). *The Illusion of Objectivity: How Language Constructs Authority*. https://doi.org/10.5281/zenodo.15395917
- Startari, A. V. (2025h). *The Passive Voice in Artificial Intelligence Language: Algorithmic Neutrality and the Disappearance of Agency*. https://doi.org/10.5281/zenodo.15464765

These works ensure traceability of terminology, formal operators, and theoretical lineage across the Startari framework. The TLOC formalizes what was previously emergent in this canon: the structural non-verifiability of obedience in generative systems.

Appendix – Methodological Corpus for Falsifiability Testing

This annex lists external sources that were consulted during the falsifiability and boundary-testing phase of this article. While none of these works are cited in the main body, their examination was essential to delineate the structural originality of the hypothesis and to contrast it with existing theoretical models.

These references helped verify that the concepts of structural substitution, grammatical execution, and algorithmic colonization of time, as developed herein, do not appear in equivalent formal or epistemological terms in prior literature.

External References Consulted

- Bender, E. M., & Koller, A. (2020). *Climbing towards NLU: On meaning, form, and understanding in the age of data.* Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.463
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). *On the dangers of stochastic parrots: Can language models be too big?* Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. https://doi.org/10.1145/3442188.3445922
- Floridi, L., & Chiriatti, M. (2020). *GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 30*, 681–694. https://doi.org/10.1007/s11023-020-09548-1
- Mitchell, M. (2023). *Artificial Intelligence: A Guide for Thinking Humans.* Penguin Books.
- Marcus, G., & Davis, E. (2020). *Rebooting AI: Building Artificial Intelligence We Can Trust.* Pantheon.

- Clune, J. (2021). *AI-GAs: Artificial Intelligence Generating Algorithms. Nature Machine Intelligence, 3*, 74–86. https://doi.org/10.1038/s42256-020-00282-1
- Chollet, F. (2019). *On the Measure of Intelligence. arXiv preprint.* https://arxiv.org/abs/1911.01547
- Turing, A. M. (1950). *Computing Machinery and Intelligence. Mind, 59*(236), 433–460.
- LeCun, Y. (2022). *A Path Towards Autonomous Machine Intelligence. Meta AI Research Whitepaper.* https://openreview.net/forum?id=BZ5a1r-kVsf
- Chomsky, N., Roberts, I., & Watumull, J. (2023). *The False Promise of ChatGPT. The New York Times.*

This annex ensures transparency regarding prior discourse while affirming that no borrowed conceptual structures, formal models, or terminologies were integrated into the theoretical body of the article. All operative constructs , including conditional substitution, trajectory opacity, and non-verifiability logic , were independently developed and tested against existing literature to establish originality and falsifiability.