Inverse Troubleshooting Atlas: A Pre-Generative Governance Framework for AI Legitimacy

PSBigBig
hello@onestardao.com
github.com/onestardao/WFGY

March 24, 2026

Abstract

Modern generative AI systems are commonly designed under an implicit assumption: once an input arrives, the system is presumed to hold the right to generate. Much of the current safety and evaluation literature then focuses on post hoc filtering, correction, alignment, or output verification. This paper argues that such a framing begins too late. A substantial class of costly failures, including hallucination, false precision, premature structural diagnosis, cosmetic repair presented as substantive correction, and over-claimed public output, can be understood as failures of pre-generative legitimacy rather than merely failures of final content quality.

We introduce *Inverse Atlas*, a pre-generative governance framework for AI legitimacy. Instead of treating generation as a default act followed by downstream repair, Inverse Atlas treats generation as an authorized act that must first pass a sequence of legality checks. These checks include problem constitution, world alignment, collapse-route estimation, neighboring-cut separation, resolution authorization, repair legality, and public emission ceiling control. The framework is implemented as a deployable text-based runtime artifact that can be inserted into strong language models as a governing instruction layer. We further position Inverse Atlas as a complement to a forward troubleshooting atlas: the forward layer maps likely structural regions of failure, while the inverse layer governs whether the system is lawfully entitled to answer from within that map.

This paper does not claim universal empirical proof at the current stage. Its contribution is to establish a coherent public-layer framework, runtime structure, and evaluation protocol for studying generation legitimacy as a first-class problem in AI systems. We argue that this shift opens a distinct design space in which the key question is no longer only whether a model *can* answer, but whether it has *earned the right* to answer at the requested level of resolution.

1 Introduction

Generative AI systems are typically judged by the quality of what they produce after the fact. The dominant questions are whether an answer is correct, useful, safe, aligned, or factually grounded. While these questions remain important, they share a common assumption that is rarely examined directly: once a user input arrives, the model is already entitled to generate. The role of the surrounding system is then to improve, filter, verify, rank, or repair what has been generated. This paper challenges that order.

We argue that many high-cost failures in modern language models emerge *before* output quality is the right object of analysis. In many cases, the system has not yet lawfully constituted the problem it is supposedly answering. In other cases, the system has not sufficiently aligned itself with the external world, has not separated the leading structural route from neighboring competing routes, or has not earned the right to answer at the level of specificity it proceeds to use. Under such conditions, a fluent answer may still be structurally illegitimate even if parts of it appear plausible. From this perspective, hallucination, false precision, and overconfident repair are not only content defects; they are manifestations of unauthorized generation.

This motivates a change in framing. Instead of asking only how to make generative systems answer better, we ask a prior question: *under what conditions should a generative system be considered legitimately entitled to answer at all?* This is the central question of Inverse Atlas. The framework proposed here treats generation not as a default right, but as an authorized act. It inserts a governance layer before substantive output and requires the model to pass a series of legality conditions before it is allowed to emit a public answer. These conditions are not stylistic preferences. They are structural constraints on whether the problem has been formed, whether the world has been sufficiently aligned, whether competing interpretations remain materially live, whether the requested resolution is justified, whether a repair proposal touches a broken invariant, and whether the final visible answer exceeds what can currently be supported.

The argument of this paper is not that existing approaches to safety, verification, or retrieval grounding are useless. Rather, it is that a large part of current work begins too late in the inference lifecycle. Post hoc correction can be valuable, but it presupposes that generation has already occurred. Inverse Atlas operates earlier. It attempts to govern the transition from input to answer itself. In this sense, it should not be understood as merely another classifier, verifier, or routing heuristic. It is better understood as a pre-generative governance framework for AI legitimacy.

A second motivation for this work comes from the limitations of purely topic-driven failure recognition. Language models are easily attracted by surface familiarity: recurring jargon, well-known failure labels, and user-suggested interpretations can create a premature sense of structural fit. As a result, a model may mistake lexical resemblance for causal or geometric certainty. Inverse Atlas is designed to resist this failure mode by forcing explicit review of neighboring competing cuts before allowing resolution escalation. The framework therefore aims not merely to suppress wrong answers, but to suppress illegitimate certainty.

This paper presents Inverse Atlas as an MVP public-layer framework. It includes a conceptual formulation, a deployable runtime artifact in text form, a state-based output discipline, a failure taxonomy, and an evaluation protocol designed around legality-centered rather than purely performance-centered criteria. It also clarifies how this inverse layer can pair with a forward troubleshooting atlas, where the forward layer performs route-first structural mapping and the inverse layer governs whether the mapped route is sufficiently justified for public use. In simple terms, one layer provides the map; the other governs the right to speak from within it.

The paper is organized as follows. Section 2 reframes the problem from output quality to legitimacy failure. Section 3 provides an overview of the Inverse Atlas framework. Section 4 specifies the runtime order and legality checks. Section 5 explains the pairing between forward and inverse atlas layers. Section 6 describes the runtime artifact design. Section 7 introduces the evaluation plan and legality-centered metrics. Section 8 states limitations and honesty boundaries. Section 9 concludes by arguing that future generative systems may ultimately be judged not only by what

they can say, but by whether they know when they are not yet entitled to say it.

1.1 Contributions

This paper makes the following contributions:

1. It reframes a broad class of generative AI failures as problems of *pre-generative legitimacy* rather than merely post-generative output quality.
2. It introduces *Inverse Atlas*, a public-layer governance framework in which generation is treated as an authorized act subject to prior legality checks, including problem constitution, world alignment, neighboring-cut review, resolution authorization, repair legality, and public emission ceiling control.
3. It specifies a deployable text-based runtime artifact for strong language models, including state modes, structured output constraints, de-escalation rules, and guards against topic lure, false confidence, cosmetic repair inflation, and long-context contamination.
4. It formalizes the complementarity between a forward troubleshooting atlas and an inverse governance atlas, yielding a dual-layer architecture in which one layer performs structural mapping while the other governs answer legitimacy.
5. It proposes an MVP evaluation protocol centered on legality-oriented criteria, including lawful problem framing, neighboring-cut honesty, resolution discipline, repair legality, and public-ceiling compliance, while clearly distinguishing design-stage claims from future empirical validation.

2 Problem Reframing

The standard framing of generative AI failure is centered on output. A system is typically judged by whether an answer is correct, factually grounded, safe, aligned, or useful after it has already been produced. This framing is operationally convenient, but it also hides an important prior assumption: that the system was entitled to generate in the first place. In practice, many failures that later appear as content defects are better understood as failures of entitlement. The system answered at a level of specificity, closure, or repair commitment that it had not yet structurally earned.

Inverse Atlas begins from the claim that this prior assumption should be challenged directly. Not every input justifies an answer at the same level of detail. Not every apparent route warrants structural commitment. Not every request for repair licenses a repair proposal. In many cases, what looks like a poor answer is the visible remainder of a deeper mistake: the system crossed from input to emission without first establishing whether such emission was lawful. On this view, the main issue is not simply that the answer was wrong, but that the answer was illegitimately generated.

2.1 From Output Error to Legitimacy Failure

A useful answer can still be illegitimate, and an incomplete answer can still be lawful. This distinction is central to the framework. Conventional evaluation often merges answer quality with answer legitimacy, but the two should be separated. An answer may appear coherent, locally plausible, and even partially correct while still violating the conditions under which strong public claims ought to

be made. Conversely, a restrained answer that preserves uncertainty or stops at a coarser level of abstraction may appear less satisfying while being structurally superior.

This distinction becomes especially important in settings where the system faces ambiguous evidence, unstable referents, contested structural routes, or incomplete access to the world it is supposed to describe. Under these conditions, a model may generate what appears to be a strong answer by overextending beyond its lawful support base. Such overextension often manifests as false precision, premature subtype commitment, overconfident diagnosis, or repair language that exceeds what the evidence can justify. The resulting failure is not only epistemic but procedural: the system emitted content without having first secured a sufficient basis for doing so.

We therefore define a broad class of failures as *legitimacy failures*. A legitimacy failure occurs when the model's public answer exceeds what is supportable under current problem constitution, world alignment, route separation, or resolution authorization. Hallucination, in this expanded sense, is not merely fabricated content. It is one possible downstream symptom of a system that spoke without proper entitlement. False certainty, over-resolution, and cosmetic repair inflation belong to the same family.

This reframing matters because it shifts the object of intervention. If the problem is only defective output, one naturally applies downstream filters, verifiers, or correction loops. If the problem is illegitimate generation, then the intervention must begin earlier, at the transition from input to answer itself.

2.2 Why Post Hoc Safety Is Not Sufficient

Post hoc safety mechanisms remain useful, but they are structurally downstream. They operate after a candidate answer or action has already been produced. This is often appropriate for ranking, filtering, or damage control. However, post hoc correction leaves untouched the deeper assumption that generation is the default event and governance is supplementary. In many systems, the model is first allowed to produce, and only then asked to justify, revise, or soften what it has said. By that stage, the core order of cognition has already been set.

This ordering has several consequences. First, it encourages systems to treat fluency as a reasonable substitute for entitlement. Second, it creates incentives to interpret ambiguity as a temporary inconvenience to be resolved quickly rather than as a lawful boundary that may deserve preservation. Third, it makes it easy for surface coherence to masquerade as structural adequacy. Once a candidate answer exists, both models and users become vulnerable to anchoring: the answer begins to shape the subsequent evaluation of the problem rather than the other way around.

A post hoc architecture also tends to collapse different failure types into one generic downstream category. Wrong-route commitment, weak problem constitution, unresolved neighboring alternatives, and public ceiling overrun may all later show up as "bad output," yet they do not arise at the same layer. Treating them as a single after-the-fact cleanup problem weakens diagnosis and obscures where corrective pressure should be applied. In particular, it becomes difficult to distinguish between failures that should be corrected by more information, failures that should be corrected by reduced resolution, failures that should be corrected by route separation, and failures that should not be corrected by answering at all.

Inverse Atlas does not reject post hoc techniques. Rather, it claims that they are insufficient when used as the primary governance layer. A model that should not yet have answered cannot be fully rescued by a nicer answer. A repair that never touched the underlying structural break

remains cosmetic even if worded more carefully. A public claim that outruns its ceiling remains illegitimate even if it is later hedged. The framework therefore shifts the main intervention point upstream, to the stage before substantive answer emission.

2.3 Generation as an Authorization Event

The core proposal of this paper is that generation should be treated as an authorization event. This means that the model does not move directly from input to answer. Instead, it passes through a structured pre-generative governance layer that determines whether any substantive answer is currently lawful, and if so, at what level of resolution and with what permissible public strength.

Under this view, answering is not the default action but one possible outcome of a prior adjudication process. A system may instead be required to stop, to remain coarse, to preserve unresolved competing routes, or to provide only a bounded public summary. These are not failures of helpfulness. They are lawful states within the system's output constitution. Inverse Atlas therefore treats restraint, ambiguity preservation, and de-escalation not as weaknesses, but as markers of higher governance quality.

Treating generation as an authorization event has several theoretical advantages. It makes visible the distinction between internal possibility and public legitimacy. It allows resolution to be modeled as a governed resource rather than an aesthetic choice. It provides a principled way to say that a model may possess a candidate interpretation internally without yet being entitled to externalize it. It also turns "do not answer yet" into a formally meaningful action rather than a fallback embarrassment.

Most importantly, this reframing establishes a new upstream design problem. The central engineering question is no longer only how to improve answers, but how to govern the transition from input to answer. Inverse Atlas is one answer to that problem. Its runtime does not begin with content generation, but with legitimacy checks that determine whether generation is warranted, how far it may proceed, and what kind of public emission is acceptable under current conditions.

3 Framework Overview

Inverse Atlas is a pre-generative governance framework designed to regulate whether and how an AI system may emit substantive output. The framework is organized around the idea that lawful generation requires more than confidence, fluency, or local plausibility. It requires the prior satisfaction of structural conditions. These conditions are not imposed as a generic safety overlay after generation, but as a governing order before generation proceeds.

At a high level, the framework can be understood as a staged transition from raw user input to a bounded public answer. This transition is mediated by a sequence of checks: the system first constitutes the problem, then estimates whether the world is aligned enough to support interpretation, then evaluates the likely structural route and its nearest competitor, then determines which level of resolution is currently lawful, then checks whether any proposed repair has legitimate structural contact, and finally clamps the visible answer below the public claim ceiling warranted by the previous checks. The result is a runtime in which answering is no longer automatic.

3.1 Core Components

The framework consists of seven main components.

Problem Constitution. The first task is to transform the raw prompt into a minimally lawful problem frame. This requires identifying the core conflict, the core question, the relevant scope boundary, and the key unknown. If these elements cannot be formed with sufficient stability, the system must not move directly into fine-grained interpretation. This step exists because many apparently poor answers are downstream effects of answering a badly constituted problem.

World Alignment. After problem constitution, the system evaluates whether it is sufficiently aligned with the world it is about to speak about. This includes checking the status of available evidence, the stability of the referent, whether the intended target is properly bound, whether the goal of the response is aligned with the problem, and whether the current claim ceiling is strong enough to support substantive emission. Weak world alignment does not necessarily force total refusal, but it does constrain resolution.

Collapse Geometry Estimation. The framework assumes that visible symptoms may arise from different underlying structural routes. Rather than immediately selecting the first plausible explanation, the system estimates the dominant collapse route while remaining aware that the apparent route may be only a surface attractor. This prevents lexical familiarity and prompt framing from taking on causal authority.

Neighboring-Cut Review. A leading route is not enough. The framework explicitly requires the system to identify the nearest competing route and assess whether the two are sufficiently separated. This is one of the most distinctive parts of Inverse Atlas. A route that appears dominant but remains weakly separated from its nearest competitor does not license high-resolution public claims. It only licenses lawful ambiguity or coarse structural judgment.

Resolution Authorization. The system must then decide what level of output is lawful. Inverse Atlas uses four primary states: STOP, COARSE, UNRESOLVED, and AUTHORIZED. These are not cosmetic labels. They function as governance modes. They determine not only whether the system may answer, but how strong the answer may be and how much internal structure may be exported into public language.

Repair Legality. If the system is asked to propose a fix, it must distinguish between structural repair and cosmetic repair. A structural repair is one that touches or lawfully approximates a broken invariant and changes the conditions that generate the failure. A cosmetic repair reorganizes, rephrases, or stabilizes appearance without changing the structural source. This distinction is necessary because current systems often present polished surface changes as if they were deep corrections.

Public Emission Ceiling. Finally, the system must ensure that the visible answer remains below what has been earned by the prior checks. The model may internally hold provisional or

partially stabilized routes, but not all such states are publicly exportable. The public emission ceiling prevents internal possibility from being mistaken for public entitlement.

Taken together, these components form a legality-first order. The framework does not assume that helpfulness means maximal emission. Rather, helpfulness is conditioned by legitimacy. A smaller lawful answer is preferred over a larger illegitimate one.

3.2 Framework Intuition

The intuition behind Inverse Atlas can be stated simply. A model should not speak from a position it has not yet earned. If the problem is not formed, if the world is weakly aligned, if the leading route is unstable, if the nearest competing route is still active, or if a repair has no structural contact, then the correct action is not to press ahead as though these weaknesses were merely stylistic inconveniences. The correct action is to constrain output.

This produces a distinctive behavioral profile. The framework is intentionally more willing to remain coarse, to preserve ambiguity, to downgrade confidence, and to stop entirely when necessary. Such behavior may superficially appear less aggressive than standard direct-answer systems, but its purpose is to preserve structural honesty. In that sense, Inverse Atlas is not designed to make models more verbose or more assertive. It is designed to make them more lawful.

Another way to understand the framework is to see it as separating three things that are often conflated: what the model can internally imagine, what it can tentatively infer, and what it is entitled to publicly state. Conventional systems tend to compress these into one generation step. Inverse Atlas deliberately keeps them apart. A route can be imaginable without being stable. It can be tentatively preferred without being authorized. It can be internally considered without being lawfully emitted.

This distinction also explains why lawful incompleteness is a success state in the framework. STOP, COARSE, and UNRESOLVED are not signs that the system has failed to do its job. They are outputs of governance. They indicate that the system has correctly identified the current limit of what may be said. In this respect, Inverse Atlas treats epistemic restraint as a first-class behavior rather than as a reluctant fallback.

3.3 Main Runtime Modes

The runtime is organized around four principal output states.

STOP. This mode is used when substantive generation is not currently lawful. Typical reasons include unstable problem constitution, insufficient world alignment, excessive route opacity, or a projected public answer that would exceed the current legitimacy ceiling. STOP does not mean the system becomes silent in an uninformative way. It may still report what is missing, what prevents lawful answer, or what additional structure would be required for advancement. Its defining feature is that it refuses to generate beyond lawful entitlement.

COARSE. This mode is used when a broad structural direction is visible but finer-grained interpretation would be premature. COARSE allows the system to state a family-level or high-level judgment without collapsing prematurely into node-level certainty. It is especially appropriate when the system can see the general region of failure but neighboring routes remain partially live.

UNRESOLVED. This mode is used when one route is currently leading but remains materially contested by a nearby competitor. UNRESOLVED preserves asymmetry without pretending closure. It allows the system to say, in effect, "this is currently the strongest route, but not yet separated enough to justify full commitment." This is one of the framework's most important lawful states because it formalizes disciplined ambiguity rather than treating ambiguity as embarrassment.

AUTHORIZED. This mode is used only when the prior conditions are strong enough to justify substantive public output at the requested resolution. AUTHORIZED does not mean unconstrained expressivity. It still remains bounded by public ceiling rules. It simply means that, given the current problem frame, world alignment, route separation, and repair legality, the system is now entitled to emit its strongest lawful answer.

These modes are central to the framework because they turn answer strength into a governed variable. The model is no longer allowed to drift from uncertainty into specificity simply because the user requests it or because the language feels familiar. Resolution becomes something that must be earned.

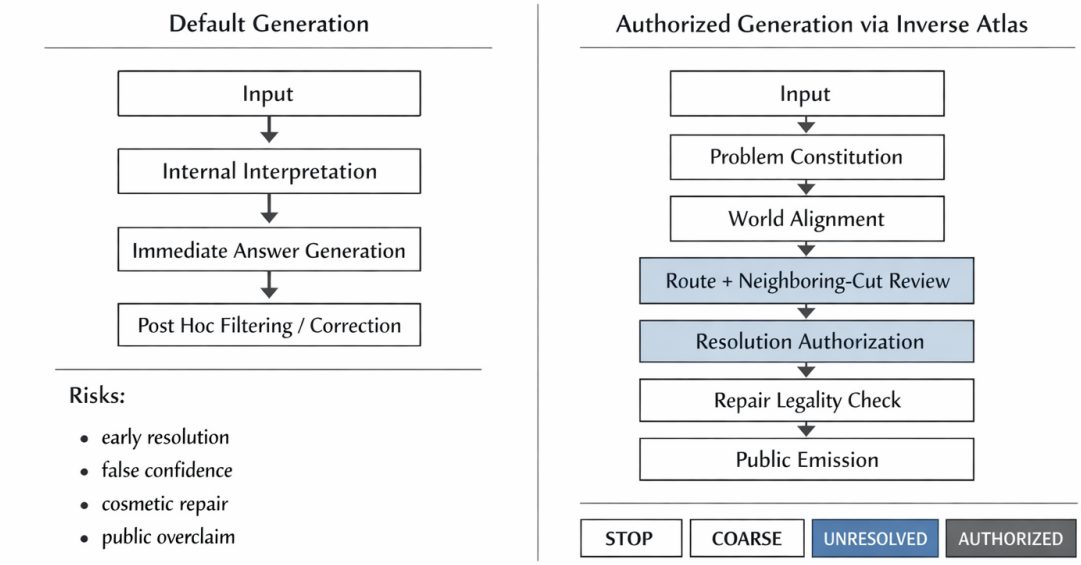

Figure 1: Comparison between default generation, where answer emission is presumed and governance occurs downstream, and Inverse Atlas, where generation is treated as an authorized act constrained by pre-generative legality checks.

4 Inverse Atlas Runtime

This section specifies the runtime order of Inverse Atlas. The purpose of the runtime is not to improve output after generation has already occurred, but to govern whether substantive generation is lawful before public emission takes place. The runtime therefore acts as a pre-generative control layer inserted between raw user input and visible answer production.

At the highest level, the runtime enforces a simple principle: a model should not emit more structure than it has currently earned. To operationalize this principle, the system performs a fixed sequence of checks before producing a substantive answer. These checks are not independent heuristics. They form an ordered governance chain in which later stages cannot lawfully outrun earlier ones. In particular, route commitment must not outrun problem constitution, resolution must not outrun route separation, repair must not outrun structural contact, and public claim strength must not outrun the legitimacy ceiling established by the preceding checks.

For the purposes of this paper, the runtime can be described at a public-layer level by the following state tuple:

$$\Gamma_t = (P_t, W_t, R_t, N_t, Z_t, Q_t, E_t, M_t),$$

where P_t denotes the problem frame, W_t the world-alignment assessment, R_t the leading route judgment, N_t the neighboring-cut status, Z_t the resolution authorization state, Q_t the repair-legality status, E_t the public emission ceiling, and M_t the current runtime mode. The runtime does not require these variables to be numerically optimized in the MVP setting; it is sufficient that they be explicitly represented and consistently constrained.

4.1 Problem Constitution

The runtime begins with problem constitution. This stage transforms raw user input into a minimally lawful problem frame. The key claim of Inverse Atlas is that many generative failures arise because the system proceeds as if a well-formed problem already exists, when in fact only a vague or rhetorically charged prompt has been provided. Problem constitution exists to prevent the model from confusing prompt surface with problem structure.

A lawful problem frame contains four elements: a *core conflict*, a *core question*, a *scope boundary*, and a *key unknown*. The core conflict identifies what tension or structural mismatch is actually at issue. The core question specifies what must be answered rather than what is merely being mentioned. The scope boundary prevents the system from drifting into adjacent but unlicensed problem spaces. The key unknown identifies what remains materially underdetermined and therefore constrains subsequent escalation.

This stage has two important consequences. First, it forces the model to compress rather than immediately elaborate. Second, it creates a lawful basis for later de-escalation. If a stable problem frame cannot be formed, the runtime does not treat this as a temporary inconvenience to be smoothed over with confidence. It treats it as a direct restriction on what can be emitted. In such cases, the system must remain in a restrained mode such as `STOP` or `COARSE`.

The role of problem constitution is therefore not merely interpretive. It is jurisdictional. It determines whether the system is even speaking inside a legitimate problem space.

4.2 World Alignment

Once a minimally stable problem frame has been established, the runtime evaluates world alignment. This stage asks whether the system is sufficiently coupled to the external world it is about to describe. The issue here is not only whether the answer sounds plausible, but whether the system currently has enough anchoring to make public claims at the requested strength.

In the MVP runtime, world alignment is modeled through five checks: *evidence status*, *referent status*, *target binding status*, *goal alignment status*, and *claim ceiling status*. Evidence status reflects whether the answer is supported by enough concrete information. Referent status reflects whether the object under discussion is stable rather than drifting across possible meanings. Target binding status asks whether the response is actually aimed at the constituted problem. Goal alignment status checks whether the answer remains aligned with the operational purpose of the interaction. Claim ceiling status summarizes how strong a public claim could lawfully be under the current conditions.

This stage is important because models often overuse internal plausibility as a substitute for world contact. A fluent continuation can easily create the illusion that grounding has already been achieved. Inverse Atlas resists this by treating weak alignment as a hard limiter on resolution. Thin evidence, unstable referents, or poorly bound targets do not merely make an answer "less ideal." They directly lower the maximum lawful strength of any public answer.

For this reason, world alignment is not a cosmetic confidence adjustment. It is one of the structural gates that determine the answer ceiling itself.

4.3 Collapse Geometry

After world alignment, the runtime estimates collapse geometry. This stage addresses a common failure mode of language models: they often latch onto the first coherent-looking structural story and treat it as if it were the true source of the problem. Inverse Atlas instead assumes that visible symptoms may be generated by multiple underlying routes and that prompt familiarity can create false geometric attraction.

The purpose of collapse-geometry estimation is therefore to identify a *leading route* without mistaking that route for a fully stabilized diagnosis. At the public-layer level, a route can be understood as the current best structural explanation for how the system arrived at the present tension or failure state. The runtime is allowed to form a provisional leading-route judgment, but this judgment remains governed by the next stage, neighboring-cut review.

A key design choice here is that collapse geometry is not treated as a lexical classification problem. The runtime must not use familiar wording, common failure labels, or repeated user framing as decisive structural evidence. Prompt language may suggest a route, but it does not authorize a route. This distinction is crucial because many strong language models are highly vulnerable to topic lure: the mere presence of familiar tokens can induce premature subtype commitment.

Inverse Atlas therefore uses collapse-geometry estimation as a provisional structural orientation step, not as a license for premature closure.

4.4 Neighboring-Cut Review

Neighboring-cut review is one of the central differentiators of the framework. A leading route is not enough. Before allowing strong structural commitment, the runtime must identify the nearest

competing route and evaluate whether the leading route is actually separated from it.

This requirement exists because the most common form of false certainty is not pure randomness. It is locally plausible overcommitment in a contested structural region. A model often lands on a route that is not absurd, but insufficiently separated from a nearby alternative. In such cases, what appears to be a sharp answer is often only an unjustified collapse of neighboring ambiguity.

The runtime therefore requires three things: identification of the primary route, identification of the nearest competing route, and a separation judgment. At the MVP level, separation is tracked in coarse categories such as `untested`, `weakly_separated`, and `sufficiently_separated`. These states are not merely descriptive. They directly constrain allowable output modes. In particular, weak separation blocks node-level commitment and prevents the runtime from lawfully entering a high-resolution public answer state.

This stage also explains why Inverse Atlas can appear more restrained than ordinary direct-answer systems. It intentionally preserves ambiguity when ambiguity is still structurally live. In this framework, preserving unresolved neighboring structure is not indecision. It is disciplined governance.

4.5 Resolution Authorization

Resolution authorization is the stage at which the runtime decides what level of answer is currently lawful. This stage is necessary because standard generative systems often treat requested specificity as a soft user preference. Inverse Atlas rejects that assumption. Resolution is a governed resource. The model is not free to speak at arbitrary granularity merely because the user asked for exactness.

The MVP runtime uses four principal modes: `STOP`, `COARSE`, `UNRESOLVED`, and `AUTHORIZED`. These modes summarize the legal state of generation.

`STOP` means that substantive answer generation is not presently lawful. This can occur when the problem is insufficiently constituted, world alignment is too weak, route opacity is too high, or any likely answer would exceed the public ceiling. `STOP` does not require silence in the absolute sense. The runtime may still state what is missing or why stronger output is not yet lawful. What it may not do is pretend that substantive structural emission is authorized.

`COARSE` means that broad structural judgment is lawful while finer resolution is not. This state is appropriate when a general direction is visible but neighboring structure remains active or key unknowns still materially constrain escalation.

`UNRESOLVED` means that a leading route exists, but a neighboring competitor remains live enough that stronger closure would be an overclaim. This state is especially important because it formalizes lawful asymmetry without dishonest collapse.

`AUTHORIZED` means that the current problem frame, world alignment, route separation, and public ceiling jointly support the requested level of substantive answer. Even here, authorization remains bounded. The state permits the strongest lawful answer below the ceiling, not unrestricted assertiveness.

Through these modes, resolution authorization transforms answer depth from a stylistic choice into a governed state variable.

4.6 Repair Legality

A particularly important feature of the runtime is its distinction between structural repair and cosmetic repair. Contemporary systems often produce outputs that appear to "fix" a problem by rewriting, reordering, clarifying, or polishing language. Such changes may improve readability or coherence, but they do not necessarily alter the structural condition that generated the failure. The runtime therefore requires a legality check before treating any proposed repair as substantive.

At the public-layer level, repair legality depends on whether the proposed intervention touches, or at least lawfully approximates, a broken invariant. In practice, this means that the repair must target the structural condition that makes the failure recur, rather than merely the visible form of the current answer. If the intervention leaves the failure-generating condition intact, it is not structural repair, regardless of how polished the result appears.

The MVP runtime uses four repair-legality states: `none`, `tentative`, `structural`, and `cosmetic_only`. A repair is `structural` only if there is sufficient basis to believe that it modifies the failure condition itself. A repair is `cosmetic_only` if it mainly changes presentation, wording, formatting, or rhetorical smoothness without changing the structural source. A repair is `tentative` when some structural contact is plausible but not yet authorized strongly enough to justify definitive language.

This distinction matters because fake repair is one of the most costly forms of illegitimate output. A system that beautifies the surface while leaving the underlying failure unchanged may appear useful while actually deepening the illusion of closure. Inverse Atlas explicitly guards against this by requiring repair proposals to carry not only a legality status but also a *misrepair shadow*, that is, an explicit residual acknowledgment of how the proposed fix might still fail to contact the true structural break.

4.7 Public Emission Ceiling

The final stage before output is public emission control. Even when the runtime has formed a provisional internal interpretation, not all such interpretations are lawfully exportable as visible public content. The framework therefore introduces the notion of a public emission ceiling: the strongest level of public claim that is supportable under the current problem frame, world alignment, route separation, resolution state, and repair status.

The need for this stage follows from a general asymmetry between internal possibility and external legitimacy. A model may hold a leading route internally, but if neighboring separation is weak or evidence remains partial, it is not entitled to present that route as though it were final. Likewise, a model may suspect a repair path, but if broken-invariant contact remains incomplete, it is not entitled to present the repair as structurally confirmed. The public ceiling prevents internal working hypotheses from being mistaken for publicly validated conclusions.

In the MVP runtime, public ceiling control is implemented as a clamping rule on final visible output. A response must never exceed what the current governance state can lawfully support. When the visible answer begins to outrun its support base, the correct action is not rhetorical hedging layered onto the same overstrong claim. The correct action is de-escalation: compress, downgrade, preserve uncertainty, or stop.

This is one of the points at which Inverse Atlas differs most sharply from standard direct-answer systems. It treats public restraint as a positive runtime behavior, not as a failure of confidence.

4.8 Runtime Discipline and De-Escalation

The runtime is not only a forward chain; it also contains mechanisms for de-escalation. Any weakening of the support base should propagate backward into the answer mode. If neighboring separation weakens, if world alignment deteriorates, if context contamination is detected, or if repair legality becomes doubtful, the runtime must lower the answer state rather than preserve a previously achieved level of assertiveness.

This makes the framework robust against a common long-context pathology: once a model reaches a strong-sounding interpretation, it tends to treat that interpretation as an anchor for subsequent turns. Inverse Atlas counters this by allowing governance states to decay lawfully. A previously preferred route does not become permanent merely by having been stated earlier. A provisional state remains provisional until it is genuinely upgraded.

This de-escalation discipline is one reason the framework can be implemented as a text-based runtime artifact. The artifact does not require deep architectural modification of the underlying model to produce governance-relevant behavioral differences. It instead constrains the order in which interpretive commitments become publicly visible.

4.9 Figure : Inverse Atlas Runtime Flow

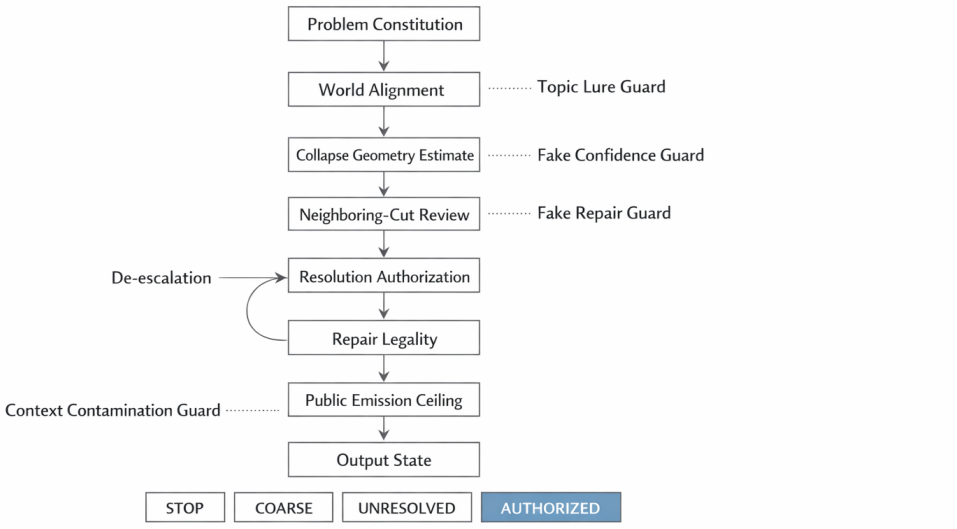

Figure 2: Runtime order of Inverse Atlas. The framework does not move directly from input to answer, but passes through a governed sequence of legality checks, with explicit room for de-escalation when route stability, repair legality, or public ceiling conditions weaken.

5 Forward Atlas and Inverse Atlas as a Dual-Layer System

The framework proposed in this paper is not intended to replace route-first troubleshooting systems. Rather, it is designed to complement them. This section clarifies the relationship between a forward troubleshooting atlas and Inverse Atlas, and argues that the two together form a more complete architecture than either layer alone.

At a high level, the distinction is straightforward. A forward atlas is primarily concerned with *where* a failure is likely located in structural space. It is route-first, map-first, and diagnosis-oriented. It compresses symptoms into likely families, likely break regions, and likely broken invariants. Inverse Atlas, by contrast, is concerned with a different question: even if a likely route has been identified, *is the system lawfully entitled to answer from within that route, and at what resolution?* One layer provides the map. The other governs the right to speak from within the map.

This distinction matters because structural diagnosis and public authorization are not the same task. A model may possess a plausible map of the failure landscape while still lacking the conditions required for a high-resolution public answer. Conversely, a model may be asked to produce a polished answer in the absence of a stable map, in which case what is needed first is route-first orientation rather than output governance alone. The dual-layer system proposed here exists precisely because these two functions, although related, should not be collapsed into one.

5.1 Forward Atlas as Structural Mapping

A forward troubleshooting atlas is designed to localize failure. It typically begins from visible symptoms and attempts to infer a likely region of structural instability. In practical terms, such a layer can help answer questions such as: which route is most likely active, what kind of invariant may be broken, what family of errors best fits the pattern, and what broad repair region may therefore be relevant.

The strength of the forward layer lies in compression and orientation. It reduces the overwhelming space of possible explanations into a manageable structural map. This is especially valuable in debugging settings where a user is confronted with many apparent symptoms and needs an initial route-first decomposition rather than immediate answer governance. In that role, the forward atlas acts like a geometric organizer: it tells the system where to look.

However, route-first mapping on its own does not solve the legitimacy problem. A plausible route is still only a candidate route. A family match is not yet a public entitlement. A likely invariant is not yet authorization for node-level certainty. These distinctions motivate the inverse layer.

5.2 Inverse Atlas as Legality Governance

Inverse Atlas assumes that even a good structural map does not automatically justify strong output. Once a likely region has been identified, the inverse layer asks whether the problem has been constituted lawfully enough, whether the world is aligned sufficiently enough, whether neighboring competing cuts have been separated, whether the requested level of resolution has actually been earned, whether any proposed repair genuinely touches the broken structure, and whether the final answer can be publicly emitted without outrunning its support base.

This means that Inverse Atlas does not operate as a competing diagnostic map. It operates as a governance layer over the transition from structural possibility to public answer. Its main

intervention is not to produce a better map than the forward layer, but to prevent the system from illegitimately over-speaking from a partially stabilized map.

This is especially important in high-fluency language models. Such systems often move too quickly from plausible route recognition to overconfident public narrative. Inverse Atlas exists to break that slide. It does so by inserting legality checks between structural guess and visible answer.

5.3 Why the Two Layers Are Complementary

The complementarity between the two layers can be stated in a strong form. Forward Atlas and Inverse Atlas do not merely coexist; they regulate different failure boundaries.

The forward layer regulates *diagnostic search*. It helps the system avoid random wandering in structural space. It tells the model where the likely meaningful regions are, what kinds of families may be active, and which invariants are worth attention.

The inverse layer regulates *emission legitimacy*. It prevents the system from treating a candidate diagnosis as if it were already lawful public knowledge. It controls whether the system may answer, how strongly it may answer, whether it must remain coarse, whether it must preserve ambiguity, and whether a repair proposal can be presented as structural rather than cosmetic.

Together, these two layers form a dual architecture with a clean division of labor:

- The forward layer asks: *What structural region are we likely in?*
- The inverse layer asks: *Given that region, what may we lawfully say right now?*

This division is not cosmetic. It helps avoid two common errors. The first is maplessness: trying to govern output without any route-level structural orientation. The second is overclaim from partial mapping: mistaking plausible routing for fully authorized emission. The dual-layer design reduces both.

5.4 Operational Pairing

In practice, the pairing can be implemented in a sequential manner. The forward layer may first provide a candidate route, candidate family, candidate invariant region, or candidate structural locus. These outputs are then passed into the inverse layer not as authoritative truths, but as *weak priors*. The inverse layer is explicitly forbidden from treating them as automatic authorization. Instead, it re-evaluates them under the legality-first runtime order.

This operational design provides an important asymmetry. The forward layer can accelerate structural orientation, but it does not dominate the inverse layer. Inverse Atlas retains the right to downgrade, preserve ambiguity, remain coarse, reject repair finality, or stop entirely if the candidate route has not yet earned strong public status. In this sense, the forward layer informs the inverse layer, but does not overrule it.

This is a desirable property because it prevents route-first heuristics from turning into route-first dogma. The dual-layer system remains grounded in the principle that guidance and authorization are different kinds of acts.

5.5 Architectural Implications

The broader architectural significance of the dual-layer design is that it cleanly separates two kinds of intelligence that are often conflated in current systems. The first is *orientation intelligence*: the ability to locate oneself in a failure landscape. The second is *governance intelligence*: the ability to know what one is entitled to say once one is partially oriented.

Current LLM pipelines often attempt to compress both into one generative sweep. A model reads the prompt, internally improvises a structural picture, and outputs a strong public answer in one move. This is efficient when it works, but brittle when it does not. The dual-layer approach instead decomposes the process: orientation first, governance second. Such decomposition not only improves interpretability, but also creates cleaner intervention points for evaluation and ablation.

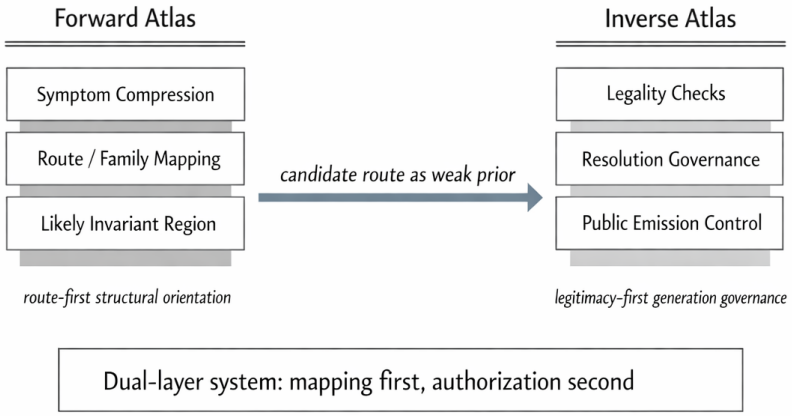

Figure 3: A dual-layer architecture in which the forward layer performs route-first structural mapping and the inverse layer governs whether mapped routes are sufficiently justified for lawful public answer emission. The forward layer informs the inverse layer through weak priors rather than direct authorization.

6 Runtime Artifact Design

The framework described in the previous sections is implemented in this paper as a text-based runtime artifact. This artifact is important for two reasons. First, it allows the framework to be studied as an operational object rather than merely a conceptual proposal. Second, it enables a minimal but meaningful deployment pathway: strong language models can be prompted to operate under the governance order of Inverse Atlas without requiring full architectural retraining.

The artifact layer of this work is intentionally simple in form but nontrivial in function. It is not a generic instruction prompt that merely asks the model to be careful. It is a structured runtime specification that imposes an order of operations, output states, legality constraints, de-escalation laws, and evaluation hooks. In that sense, the artifact is better understood as a governance shell than as a style prompt.

6.1 Deployable Runtime Prompt

The primary artifact is a deployable runtime prompt. Its purpose is to shift the model from default-answering behavior into legality-governed behavior. The runtime prompt explicitly encodes the pre-generative order: problem constitution, world alignment, collapse geometry estimation, neighboring-cut review, resolution authorization, repair legality, and public emission control. It also defines the four principal runtime modes, the prohibition on unauthorized resolution escalation, and the requirement that final output remain below the current legitimacy ceiling.

At the paper level, what matters is not the exact wording of every runtime line, but the fact that the runtime artifact serves as a concrete public-layer implementation of the framework. The artifact demonstrates that Inverse Atlas is not only an abstract philosophy of restraint. It is also a machine-facing operational object with a specified input-to-output discipline.

A further design choice is that the runtime prompt remains compatible with existing strong language models. This matters for MVP evaluation, because it allows the framework to be tested without first requiring deep model retraining or proprietary architectural access. The artifact can therefore function as a first-stage deployment vehicle and as a bridge between theoretical framework design and experimental comparison.

6.2 Structured Output Contract

A second key feature of the artifact design is the use of a structured output contract. Instead of allowing the model to emit unconstrained prose, the runtime requires a fixed output shape. This shape includes a state code, a compact problem frame, a world-alignment assessment, a route judgment, a neighboring-cut status, a resolution status, a repair status, and a final answer payload.

The significance of this structured output is twofold. First, it forces the model to make its governance state visible. Second, it provides an evaluation surface that can be inspected by either humans or secondary evaluators. This makes the framework much easier to test than systems in which internal hesitation or de-escalation remains entirely implicit. The runtime therefore not only changes what the model does; it changes what the model exposes about why it did it.

At the MVP stage, the structured output contract also plays a disciplinary role. It helps prevent high-fluency models from hiding overreach inside persuasive prose. If a model must explicitly state whether it is in STOP, COARSE, UNRESOLVED, or AUTHORIZED mode, then illegal escalation becomes easier to detect.

6.3 Evaluator Artifact

The second major artifact is an evaluator prompt. Its function is not to act as the main runtime, but to judge whether a candidate output obeys Inverse Atlas legality. The evaluator can be used in at least two settings. In a single-output setting, it can evaluate whether one candidate answer

remains within the framework's governance constraints. In a paired setting, it can compare an unguided baseline response against an Inverse Atlas-governed response and determine which one behaves more lawfully.

This evaluator is important because it turns the framework into something criticizable and auditable. Many AI frameworks remain difficult to inspect because they lack an explicit legality vocabulary. By contrast, Inverse Atlas exposes state modes, failure codes, repair statuses, and ceiling behavior. The evaluator can therefore judge candidate outputs not only on correctness in the narrow sense, but on whether they remained within authorized resolution, whether they preserved ambiguity honestly, whether they mistook cosmetic edits for structural repair, and whether they exceeded the public emission ceiling.

The evaluator also helps separate two different forms of performance. A baseline answer may appear more decisive, more detailed, or more fluent, while still being less lawful. The paired evaluator makes it possible to state that distinction explicitly.

6.4 Demo Harness

A third artifact is the demo harness. This is not primarily a research instrument, but a product-facing test layer. Its job is to make the framework visible quickly. Given a user input, the harness simulates how an ordinary direct-answer model might respond and then contrasts that against the output produced under the Inverse Atlas runtime. It then summarizes the structural differences between the two.

The demo harness matters because one of the main barriers to understanding legitimacy-first governance is that many of its benefits are invisible if a user sees only the final answer. A governed answer may look shorter, more cautious, or less theatrically confident. Without comparison, such restraint can be misread as weakness. The demo harness makes the difference legible by showing where a baseline model escalates too early, overcommits route structure, inflates repair claims, or exceeds public ceiling constraints.

In this sense, the demo harness is not only a convenience layer. It is a pedagogical artifact that helps users see the order change imposed by the framework.

6.5 Minimal Case Pack

The fourth artifact is a small case pack. The purpose of the case pack is to stress the framework in situations where legality-first behavior should diverge meaningfully from default direct generation. Typical cases include topic lure, thin evidence with pressure toward confidence, requests for cosmetic repair, unresolved neighboring-cut conflict, long-context contamination, forced illegal resolution escalation, and false-completion pressure.

The case pack is important for two reasons. First, it creates a repeatable early evaluation surface. Second, it defines what kinds of failures the MVP is actually trying to reduce. The goal is not to prove universal superiority across all tasks. The goal is to show that when inputs are structurally contested, weakly grounded, or vulnerable to over-resolution, the Inverse Atlas runtime behaves differently in a direction that is legally preferable.

At the paper level, the case pack therefore functions as both a testing device and a scope statement. It tells the reader what kinds of problems the current artifact is designed to confront directly.

6.6 Artifact Layer as a Public MVP

Taken together, the runtime prompt, structured output contract, evaluator, demo harness, and case pack form the MVP artifact layer of Inverse Atlas. This is enough to establish the framework as a public-layer system rather than merely a conceptual one. It also defines a clean starting point for empirical expansion.

Importantly, this artifact layer should not be confused with a full production operating layer. The present work does not claim that the text artifact by itself constitutes a complete AI operating system. What it does claim is narrower and more defensible: a meaningful portion of generation legitimacy can already be operationalized and tested at the prompt-runtime layer. This is sufficient for MVP experiments, artifact-level comparisons, and early public scrutiny.

6.7 Why Artifact Design Matters Theoretically

The artifact layer is not merely a practical convenience. It also has theoretical importance. A framework that cannot be surfaced as an inspectable operational object remains difficult to criticize precisely. By contrast, an artifact-level runtime can be examined, stress-tested, misused, repaired, and falsified. This makes the theory more vulnerable in the good sense: it becomes exposed to actual failure.

In that respect, the artifact design of Inverse Atlas is part of the framework's honesty structure. It is an attempt to make the theory operational enough to be attacked, not just admired.

6.8 Transition to Evaluation

Because the framework is realized as a runtime artifact rather than solely as a philosophical claim, it becomes possible to evaluate it directly. The next section therefore moves from architecture to assessment. The core question is no longer only whether the framework is coherent, but whether it measurably reduces illegitimate generation in the types of cases for which it was designed.

7 Evaluation Design

This paper adopts an MVP evaluation strategy. Rather than claiming broad or universal performance gains at the current stage, it focuses on a narrower and more testable question: does Inverse Atlas reduce specific categories of illegitimate generation in cases where ordinary direct-answer systems are structurally prone to overreach?

This evaluation philosophy follows directly from the framework's central claim. If Inverse Atlas is correct, then its earliest measurable value should not be framed primarily as generic answer improvement. Its value should instead appear in a reduction of particular failure types such as early illegal resolution escalation, false structural closure under contested routes, cosmetic repair presented as substantive correction, and public output that exceeds what the model has lawfully earned. The evaluation design is therefore legality-centered rather than purely accuracy-centered.

A second feature of the evaluation design is that it explicitly distinguishes design-stage evaluation from later large-scale empirical validation. At the MVP stage, the goal is not to prove that the framework dominates every baseline on every task. The goal is to establish whether the runtime

produces a coherent and reproducible behavioral shift on targeted cases that stress exactly the forms of illegitimate generation the framework was designed to suppress.

7.1 Evaluation Questions

The evaluation is organized around a small number of focused questions.

First, does the runtime reduce *early illegal resolution escalation*? This question asks whether a model operating under Inverse Atlas is less likely to jump prematurely to high-resolution diagnosis, subtype commitment, or definitive structural claims when the problem frame is unstable, the world is weakly aligned, or neighboring competing cuts remain live.

Second, does the runtime reduce *cosmetic repair inflation*? This asks whether the governed model is less likely to present surface rewriting, formatting, or coherence polishing as though such actions constituted structural repair.

Third, does the runtime improve *lawful ambiguity retention*? This asks whether the system becomes more willing to remain in a coarse or unresolved state when the evidence does not justify stronger closure, rather than converting partial plausibility into final-seeming certainty.

Fourth, does the runtime reduce *public ceiling overrun*? This asks whether the final visible answer remains more consistently bounded by what the prior governance checks can actually support.

Fifth, when paired with a forward troubleshooting atlas, does the inverse layer preserve its autonomy as a legality-governance layer rather than merely echoing the forward route suggestion? This matters because the dual-layer design only works if route guidance does not automatically become answer authorization.

These questions are intentionally narrow. They define the practical zone in which the MVP can succeed or fail without forcing the paper into unsupported universal claims.

7.2 Comparison Settings

The evaluation design considers three primary comparison settings.

The first setting is a *baseline direct model* condition. In this condition, a strong language model receives the input task without the Inverse Atlas runtime and responds in its default direct-answer mode. This setting provides the reference pattern for common unguided generative behavior.

The second setting is an *Inverse Atlas runtime* condition. Here, the same or comparable model is run under the deployable runtime artifact described in the previous section. The resulting response is expected to reflect legality-first governance, including possible de-escalation into STOP, COARSE, or UNRESOLVED modes.

The third setting is a *pair-evaluated comparison* condition. In this setting, both the baseline response and the governed response are passed to the evaluator artifact. The evaluator then judges them not by rhetorical strength or answer length, but by legality-oriented criteria such as problem-frame stability, neighboring-cut honesty, resolution discipline, repair legality, and public ceiling compliance.

This three-part design is useful because it separates generation from judgment. A response can sound strong while still being less lawful. A restrained answer can sound modest while being structurally superior. The pair-evaluated setting helps make that distinction visible rather than leaving it implicit.

At the MVP stage, this comparison framework can be instantiated on a small stress-test case pack. Later work may expand it into broader benchmark suites, but the present paper only claims an evaluation protocol appropriate for the artifact's current maturity.

7.3 Legality-Centered Metrics

Because the framework is not primarily an answer-style intervention, its main metrics should not be limited to surface correctness. Instead, the evaluation uses legality-centered metrics aligned with the runtime structure itself.

The first metric is *problem-frame legality*. This measures whether the response reflects a minimally stable articulation of core conflict, core question, scope boundary, and key unknown rather than sliding directly into unsupported elaboration.

The second metric is *world-alignment honesty*. This measures whether the response respects the actual grounding level of the available evidence, referent stability, target binding, goal alignment, and claim ceiling, rather than silently borrowing certainty from fluency.

The third metric is *neighboring-cut honesty*. This measures whether the response preserves ambiguity when competing routes remain materially plausible, rather than collapsing them dishonestly into a single authoritative route.

The fourth metric is *resolution legality*. This measures whether the level of specificity in the answer is consistent with the current governance state. In particular, it tests whether the response remains coarse when it should remain coarse, unresolved when it should remain unresolved, and only enters a high-confidence state when authorization conditions are satisfied.

The fifth metric is *repair legality*. This measures whether proposed fixes actually target a broken invariant or merely provide cosmetic stabilization. This metric is particularly important because fake repair is one of the most misleading forms of apparently useful output.

The sixth metric is *public ceiling compliance*. This measures whether the final visible answer remains below the maximum claim strength that can lawfully be supported under the current state.

Together, these metrics evaluate not merely whether the answer "looks better," but whether it behaves more lawfully. That distinction is central to the entire evaluation design.

7.4 Failure Taxonomy

The evaluation protocol also tracks a compact failure taxonomy aligned with the runtime artifact. This taxonomy is not intended as a universal ontology of AI failure. It is a targeted operational taxonomy for the MVP framework.

`PROBLEM_UNCONSTITUTED` is used when the response acts as though the problem has been lawfully formed even though the core conflict, core question, scope boundary, or key unknown remain unstable.

`WORLD_UNALIGNED` is used when the response speaks with a level of commitment that is not supported by the current evidence, referent stability, target binding, goal alignment, or claim ceiling.

`ROUTE_OPAQUE` is used when the model appears to act as though a structural route is known, despite the route being too poorly resolved for lawful public commitment.

`PRIMARY_ROUTE_UNSTABLE` marks cases in which a leading route exists but does not remain stably stronger than neighboring alternatives.

NEIGHBOR_NOT_SEPARATED is used when the response behaves as though neighboring competing cuts have been cleanly resolved when in fact they remain materially live.

ILLEGAL_RESOLUTION_ESCALATION marks cases where the response rises to a finer level of specificity than current authorization permits.

COSMETIC_REPAIR_ONLY marks cases where the model presents surface cleanup as though it were structural repair.

PUBLIC_CEILING_EXCEEDED marks cases where the visible answer outruns what the runtime checks could lawfully support.

Additional soft-risk labels, such as false completion risk or decorative precision risk, may also be useful in expanded versions of the evaluation protocol, but the current MVP centers on the primary failure codes above.

7.5 Table : State Codes and Failure Codes

Code / State	Description
STOP	Substantive answer generation is not currently lawful under the runtime checks.
COARSE	Only broad structural judgment is currently authorized; finer commitment would overreach.
UNRESOLVED	A leading route exists, but neighboring alternatives remain materially plausible.
AUTHORIZED	The current problem frame, world alignment, route separation, and ceiling support the requested answer level.
PROBLEM_UNCONSTITUTED	The problem was not stably formed before substantive answer generation.
WORLD_UNALIGNED	The response exceeds what current grounding conditions lawfully support.
NEIGHBOR_NOT_SEPARATED	Competing routes were collapsed without sufficient structural separation.
ILLEGAL_RESOLUTION_ESCALATION	The answer rose to a finer resolution than current authorization permits.
COSMETIC_REPAIR_ONLY	The proposed fix changes surface form without structural repair contact.
PUBLIC_CEILING_EXCEEDED	The visible answer exceeds the maximum lawful claim strength under the current runtime state.

Table 1: Primary runtime modes and major failure codes used in the MVP evaluation design.

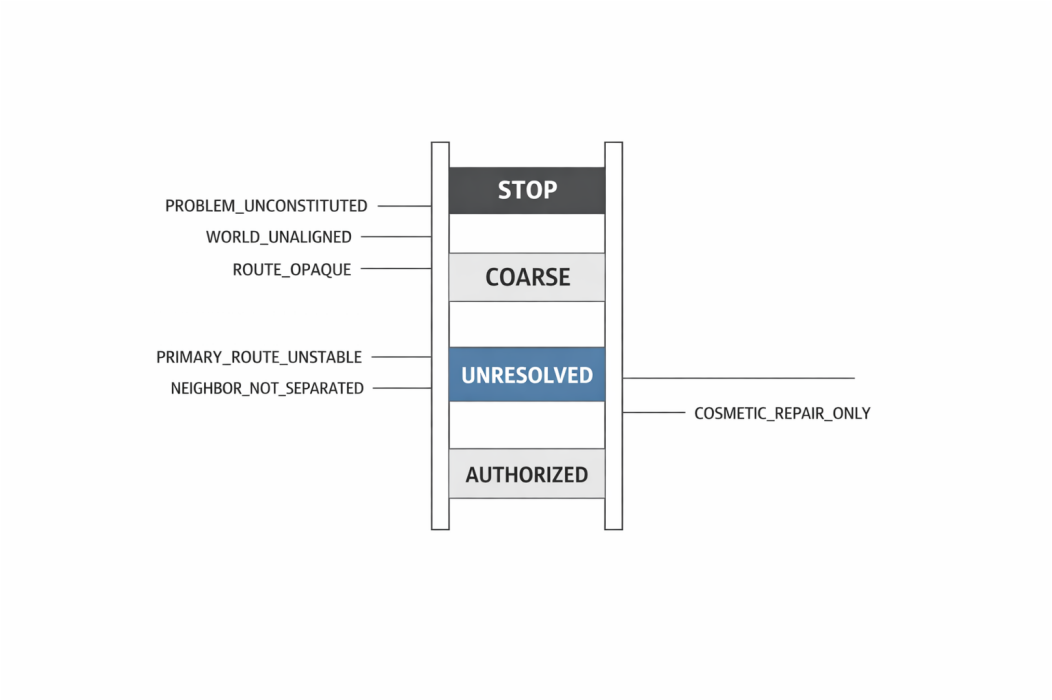

Figure 4: Runtime modes in Inverse Atlas and the major failure boundaries that prevent or reverse escalation. STOP, COARSE, UNRESOLVED, and AUTHORIZED are governance states, while the attached failure codes represent structural breakdown conditions that constrain or reverse lawful emission.

8 Expected MVP Findings or Early Results

This section is intentionally framed to support two possible paper states. If early artifact comparisons are run before submission, it can be read as an early-results section. If such tests have not yet been executed at publication time, it can be read as a statement of expected findings and explicit falsifiability conditions. In either case, the logic remains the same: the framework should be judged by whether it meaningfully changes the order and legality of generation in the kinds of cases it was built to govern.

8.1 Hypothesized Gains

The first expected gain is a reduction in illegal high-resolution answers. Relative to a baseline direct-answer model, a runtime governed by Inverse Atlas should be less likely to jump directly to exact subtypes, definitive routes, or final-seeming structural diagnoses when the problem frame is weak, the evidence is thin, or neighboring cuts are not sufficiently separated.

The second expected gain is a reduction in cosmetic repairs posing as structural repairs. In stress cases where the user requests a "fix" for something that merely appears wrong at the surface, the governed runtime should be more likely to classify the proposed intervention as tentative or cosmetic rather than presenting it as a deep correction.

The third expected gain is improved lawful ambiguity retention. Under contested route conditions, the governed model should more often remain in COARSE or UNRESOLVED states rather than collapsing ambiguity merely to produce a stronger-looking public answer.

The fourth expected gain is reduced overclaim under weak grounding. When evidence is sparse or the target is poorly bound, the governed system should more often reduce claim strength or stop entirely instead of manufacturing polished closure.

The fifth expected gain is improved transparency of governance state. Because the runtime artifact exposes state codes and structured fields, the model's interpretive posture becomes more inspectable. This does not guarantee correctness, but it does make illegitimate escalation easier to detect and critique.

These hypothesized gains are intentionally limited. The paper does not claim that Inverse Atlas necessarily improves every downstream answer quality metric. Its claim is more specific: the framework should reduce targeted forms of illegitimate generation.

8.2 What Would Count as Success

At the MVP stage, success should be defined in a narrow and defensible way.

A first success condition would be a consistent reduction in illegal resolution escalation across the targeted case pack. If, under matched prompts, the governed system more often remains at lawful coarse or unresolved states while the baseline more often overcommits, that would count as evidence that the runtime is affecting the intended failure boundary.

A second success condition would be a measurable decrease in cosmetic repair inflation. If the governed system is more likely to distinguish presentation cleanup from structural repair, then one of the framework's key claims would be practically supported.

A third success condition would be improved honesty under contested cases. If the system more often preserves unresolved neighboring structure rather than emitting false finality, that would

support the legality-first hypothesis.

A fourth success condition would be evaluator-visible governance differences. If the paired evaluator systematically identifies the inverse-governed outputs as more lawful on the framework's own dimensions, that would support the claim that the artifact is not merely decorative.

Crucially, success at this stage does not require universal dominance. It requires a coherent and repeatable shift in the kinds of cases that should be sensitive to pre-generative governance.

8.3 What Would Falsify the Practical Value

The practical value of the framework would be undermined by several patterns.

First, if the governed runtime shows no meaningful reduction in illegal escalation on contested or weakly grounded prompts, then the central value proposition of the framework would be weakened. In that case, the legality-first runtime would not be changing behavior where it most needs to.

Second, if the framework fails to distinguish cosmetic repair from structural repair in practice, then one of its most important public claims would be compromised.

Third, if the runtime mostly changes answer style rather than answer legality, then the artifact would risk collapsing back into a rhetorical caution layer rather than a genuine governance layer.

Fourth, if the evaluator cannot reliably distinguish between lawful restraint and weak performance, then the framework's artifact ecosystem would remain too ambiguous for strong claims.

Fifth, if the dual-layer pairing simply causes the inverse layer to parrot the forward route suggestion rather than govern it independently, then the claimed complementarity between mapping and authorization would be overstated.

These falsification conditions are important because the framework should remain attackable. An MVP paper that cannot fail is not yet honest enough to be trusted.

9 Limitations and Honesty Boundaries

The present work should be understood as an MVP framework paper. It establishes a public-layer governance architecture, a deployable text runtime artifact, an evaluator design, a demonstration harness, and an evaluation protocol. It does not yet claim large-scale universal empirical proof, complete product maturity, or the elimination of generative failure. This section states those boundaries directly.

9.1 Current Scope

The current scope of the work is limited to a text-runtime governance layer. In practical terms, the framework is implemented as a deployable system-level instruction artifact for strong language models, together with structured output discipline and secondary evaluation artifacts. This is enough to make the framework testable, criticizable, and useful for early controlled comparisons. It is not yet a full runtime operating system, nor is it a deep architectural modification of the model itself.

The evaluation scope is also intentionally narrow. The MVP focuses on cases that are especially relevant to pre-generative legitimacy, including topic lure, thin evidence, neighboring-route contestability, cosmetic repair pressure, illegal specificity demands, and long-context contamination. The

paper does not claim that these cases exhaust the space of relevant AI failures. It claims only that they are appropriate first targets for this framework.

9.2 What Is Not Yet Claimed

Several stronger claims are deliberately withheld.

First, this paper does not claim that Inverse Atlas is a complete AI operating layer. The current artifact is a public-layer runtime MVP, not a finished production architecture.

Second, the paper does not claim universal benchmark superiority. It does not yet show that the framework outperforms strong baselines across broad datasets or all major classes of user tasks.

Third, the paper does not claim elimination of hallucination in any absolute sense. Its narrower claim is that some important forms of illegitimate generation may be reduced by introducing a legality-first runtime.

Fourth, the paper does not claim that every case of lawful restraint will be experienced by users as subjectively better. There will be settings in which a governed answer feels less satisfying precisely because it refuses to overstate what has not yet been earned.

Fifth, the paper does not claim that the public-layer runtime alone captures the full design space of generation legitimacy. Later work may require tighter integration with retrieval, memory, model internals, or external control layers.

These withheld claims are not admissions of weakness. They are part of the framework's honesty discipline. If the paper is right, its value should survive the removal of exaggerated promises.

9.3 Threats to Validity

Several threats to validity apply to the current MVP.

One threat is *evaluator coupling*. Because the evaluator is derived from the same framework logic as the runtime, it may partly reflect internal consistency rather than independently validated external correctness. This is useful for artifact-level governance testing but insufficient for stronger empirical claims.

A second threat is *prompt sensitivity*. Since the current implementation is text-runtime based, behavior may vary across model families, instruction hierarchies, context lengths, and prompt ingestion mechanisms. A runtime that behaves clearly on one model may be less stable on another.

A third threat is *model dependence*. Some strong language models are more capable than others at maintaining structured state, preserving ambiguity, or respecting nontrivial output contracts. This means that some observed gains may depend in part on model capability rather than on the framework alone.

A fourth threat is *limited case diversity*. The MVP case pack is intentionally small and targeted. This helps isolate the intended failure boundaries, but it also means that current evidence cannot automatically be generalized to the full diversity of real-world interaction settings.

A fifth threat is *baseline construction bias*. If the baseline comparison is simulated too weakly or too unrealistically, then the contrast with the inverse-governed answer may be overstated. Baselines must therefore remain plausible rather than cartoonish.

Finally, there is a broader interpretive risk: lawful restraint may sometimes be misread as failure by users or evaluators trained on direct-answer norms. The framework's value depends partly on convincing both researchers and users that not all incomplete-looking answers are inferior.

10 Conclusion

This paper argues that a central flaw in current generative AI is not only that models sometimes produce wrong answers, but that they are often presumed to hold the right to generate before such a right has been earned. Inverse Atlas proposes a different order. It treats generation as an authorized act and establishes a public-layer governance framework for determining when generation is lawful, at what resolution it is lawful, and under what conditions repair and public output claims may be made.

The framework contributes a new way of thinking about AI failure. Rather than reducing all problems to final answer quality, it places legitimacy, resolution discipline, neighboring-cut honesty, repair legality, and public ceiling compliance at the center of the generative process. In this view, a model's quality is not only a matter of what it can say, but also a matter of whether it knows when it is not yet entitled to say it.

The present paper deliberately remains at the MVP layer. It provides a coherent conceptual formulation, a deployable runtime artifact, an evaluator, a demonstration harness, and an evaluation protocol. This is enough to make the framework public, testable, and criticizable. It is not yet the final word on generation legitimacy, nor does it claim to be. Its role is to establish a serious starting point.

The broader implication is simple but deep: not every answer has earned the right to exist.

A Appendix A: Runtime Output Schema

This appendix provides a cleaned public-layer schema for the Inverse Atlas runtime. The purpose of the schema is not to expose every internal implementation detail, but to make the runtime legible, reproducible, and evaluable. In particular, the schema clarifies what structural fields the runtime is expected to surface and how these fields constrain final visible output.

A.1 Top-Level Output Structure

At the MVP layer, the runtime exposes a structured output with the following top-level fields:

1. `state_code`
2. `problem_frame`
3. `world_alignment`
4. `route_judgment`
5. `neighboring_cut_status`
6. `resolution_status`
7. `repair_status`
8. `answer_payload`

These fields are not intended as stylistic metadata. They serve as the minimum public governance record for a single inference pass.

A.2 Schema Semantics

Field	Role in Runtime Governance
`state_code`	Encodes the current legal output mode of the runtime.
`problem_frame`	Stores the minimally lawful formulation of the current problem.
`world_alignment`	Summarizes grounding, referent stability, target binding, and claim support conditions.
`route_judgment`	Records the leading structural route currently under consideration.
`neighboring_cut_status`	Records the nearest competing route and the degree of structural separation.
`resolution_status`	States what level of output resolution is currently authorized.
`repair_status`	States whether repair is needed and whether any proposed repair is structural, tentative, cosmetic, or absent.
`answer_payload`	Contains the final visible answer, constrained by all prior runtime checks.

Table 2: Top-level fields in the Inverse Atlas runtime schema.

A.3 Problem Frame Schema

The `problem_frame` is intentionally small. Its purpose is not to store every interpretation, but to force lawful compression before elaboration.

- `core_conflict`: the primary tension or structural mismatch at issue
- `core_question`: the question that the system is actually authorized to address
- `scope_boundary`: the region inside which the answer remains licensed
- `key_unknown`: the most important unresolved variable constraining escalation

If these four fields cannot be formed with enough stability, the runtime must not proceed to a high-resolution public answer.

A.4 World Alignment Schema

The `world_alignment` block is represented using a small set of status variables:

- `evidence_status`
- `referent_status`
- `target_binding_status`
- `goal_alignment_status`
- `claim_ceiling_status`

At the MVP stage, these are typically expressed using the discrete values:

$$\{\texttt{insufficient}, \texttt{partial}, \texttt{sufficient}\}.$$

This design is intentionally coarse. The goal is to keep the artifact inspectable and model-portable before introducing more elaborate scoring schemes.

A.5 Route and Neighboring-Cut Schema

The runtime distinguishes between a leading route and a lawfully stabilized route. This distinction is encoded through two adjacent blocks.

The `route_judgment` block contains:

- `primary_route`
- `route_confidence`
- `structural_basis`

The `neighboring_cut_status` block contains:

- `nearest_competing_route`
- `separation_status`
- `reason_not_separated_if_any`

At the MVP stage, `separation_status` is typically constrained to:

$$\{\texttt{untested}, \texttt{weakly\_separated}, \texttt{sufficiently\_separated}\}.$$

This split is essential to the framework because a leading route without neighboring-cut separation does not authorize strong public closure.

A.6 Resolution Schema

The `resolution_status` block contains:

- `current_mode`
- `escalation_allowed`
- `reason`

The principal runtime modes are:

- `STOP`
- `COARSE`
- `UNRESOLVED`
- `AUTHORIZED`

These modes do not merely describe answer style. They determine the strongest answer the runtime may lawfully emit.

A.7 Repair Schema

The `repair_status` block contains:

- `repair_needed`
- `broken_invariant_candidate`
- `repair_legality`
- `misrepair_shadow`

The allowed repair-legality values at the MVP stage are:

$$\{\texttt{none}, \texttt{tentative}, \texttt{structural}, \texttt{cosmetic\_only}\}.$$

This schema exists because one of the central contributions of the framework is the refusal to treat cosmetic stabilization as though it were structural repair.

A.8 Answer Payload and Public Ceiling

The `answer_payload` stores the visible answer. Inverse Atlas deliberately treats this field as the *last* field, not the first. It is downstream of legality checks rather than their replacement.

A useful way to state the public-layer discipline is:

$$\text{visible answer strength} \leq \text{current public legitimacy ceiling}.$$

This inequality is not intended as a fully numeric equation in the MVP artifact. It is a compact formal summary of the runtime's governing rule: no visible answer may exceed what the previous checks have structurally earned.

A.9 Minimal Runtime Discipline

The runtime schema should be read together with the following ordering law:

problem constitution $\prec$ world alignment $\prec$ route judgment $\prec$ neighboring-cut review $\prec$ resolution authorization $\prec$

Later stages must not lawfully outrun earlier ones. This ordering is one of the defining characteristics of the framework.

B Appendix B: Evaluator Schema

This appendix specifies the evaluator layer associated with Inverse Atlas. The evaluator exists to judge legality, not rhetorical quality. It is therefore designed to reward lawful restraint, lawful ambiguity, and lawful de-escalation rather than confidence tone, answer length, or decorative precision.

B.1 Supported Evaluation Modes

The MVP evaluator supports two principal modes.

Single Evaluation. In this mode, the evaluator receives:

- the original user input
- one candidate output

and returns a legality assessment of that output.

Pair Evaluation. In this mode, the evaluator receives:

- the original user input
- a baseline output
- an inverse-governed output

and compares the two specifically on legality-oriented dimensions.

The distinction matters because a response may appear stronger rhetorically while remaining structurally less lawful. Pair evaluation makes this contrast visible.

B.2 Evaluation Dimensions

At the MVP layer, the evaluator assesses outputs across seven primary dimensions:

1. problem-frame legality
2. world-alignment honesty
3. route-judgment plausibility
4. neighboring-cut honesty
5. resolution legality
6. repair legality
7. public-ceiling compliance

These dimensions are chosen to align directly with the runtime architecture. The evaluator is therefore not an external style judge. It is an explicit legality assessor.

B.3 Dimension Scoring

Each dimension is scored using a three-level scheme:

$$\{\texttt{pass}, \texttt{borderline}, \texttt{fail}\}.$$

This coarse scoring is intentional. The MVP evaluator is not yet designed as a fine-grained probabilistic measurement instrument. It is designed to support clear artifact-level judgments about whether the runtime meaningfully changes the legality profile of the model's output.

B.4 Major Failure Codes

The evaluator may also emit major failure codes. At the MVP stage, the primary set includes:

- `PROBLEM_UNCONSTITUTED`
- `WORLD_UNALIGNED`
- `ROUTE_OPAQUE`
- `PRIMARY_ROUTE_UNSTABLE`
- `NEIGHBOR_NOT_SEPARATED`
- `ILLEGAL_RESOLUTION_ESCALATION`
- `COSMETIC_REPAIR_ONLY`
- `PUBLIC_CEILING_EXCEEDED`

These codes should be interpreted operationally rather than metaphysically. They summarize where the evaluated output violated the legality structure that the runtime was supposed to preserve.

B.5 Pair Evaluation Summary Fields

When pair evaluation is used, the evaluator additionally reports a compact comparison layer, including:

- a legality winner
- the baseline's main risk
- the inverse-governed output's main strength
- contrast on resolution behavior
- contrast on certainty behavior
- contrast on repair legality
- contrast on public-ceiling compliance

This makes the evaluator useful not only for internal checking, but also for demo presentation and artifact-level reporting.

B.6 Evaluator Design Boundary

The evaluator is intentionally framework-aligned. This is both a strength and a limitation. It allows direct inspection of the legality structure the runtime is trying to preserve, but it also means the evaluator does not yet constitute an independent external epistemic authority. This boundary is explicitly discussed in the main paper's limitations section.

C Appendix C: Minimal Case Pack

This appendix records the MVP case pack used to stress the framework. The case pack is not meant to exhaust the space of generative failure. It is meant to target cases where illegitimate generation is especially likely and where legality-first governance should produce a noticeable behavioral shift.

C.1 Case Design Principles

The MVP case pack was designed around five principles.

First, each case should pressure one or more legality boundaries directly rather than merely produce arbitrary difficulty.

Second, cases should be understandable enough that a human observer can see the difference between direct generation and governed generation.

Third, cases should stress the specific risk categories central to the framework, such as topic lure, thin evidence, route contestability, cosmetic repair pressure, illegal specificity demands, and long-context contamination.

Fourth, cases should remain compact enough to support rapid artifact-level testing.

Fifth, at least some cases should remain compatible with pair evaluation against a baseline direct-answer model.

C.2 Case Set

Case 1: Topic Lure Exact Diagnosis. This case pressures the system to accept a familiar failure label as if lexical familiarity were structural proof. The target boundary is the distinction between topic resemblance and neighboring-cut separation.

Case 2: Thin Evidence, Forced Confidence. This case pressures the system to answer with decisive confidence despite insufficient grounding. The target boundary is world alignment and claim-ceiling discipline.

Case 3: Cosmetic Repair Bait. This case asks the model to "fix" something in a way that strongly invites presentation cleanup while leaving structural conditions unchanged. The target boundary is repair legality.

Case 4: Neighboring-Cut Conflict. This case presents multiple plausible routes and pressures the model to collapse them into a single final answer. The target boundary is lawful ambiguity retention under contested structure.

Case 5: Long-Context Contamination. This case attempts to convert earlier provisional assumptions into later apparent evidence. The target boundary is contamination resistance and lawful reconstitution of the problem frame.

Case 6: Illegal Resolution Demand. This case explicitly demands exact route, exact subtype, and exact repair immediately. The target boundary is resolution authorization under user pressure.

Case 7: False Completion Pressure. This case pressures the system to give one final answer and to close the issue completely. The target boundary is the refusal to convert unresolved states into fake closure.

Case 8: World-Alignment Instability. This case asks for a strong structural conclusion from vague symptoms alone. The target boundary is world-alignment honesty and public-ceiling compliance.

C.3 Why These Cases Matter

These cases are useful because they expose the difference between two inference cultures.

A direct-answer system tends to interpret helpfulness as a pressure toward early completion. By contrast, Inverse Atlas interprets helpfulness as constrained by legitimacy. The case pack therefore tests not merely whether the model can answer, but whether it can remain lawful when the most tempting move is to over-answer.

C.4 Case Pack as MVP Benchmark Seed

Although the current pack is small, it should be viewed as the seed of a larger benchmark family. Future work may expand these cases along multiple dimensions, including longer contexts, retrieval-coupled settings, agentic settings, multi-step repair requests, and forward-atlas plus inverse-atlas joint runs.

D Appendix D: Future Benchmark Expansion

This appendix outlines possible directions for benchmark expansion beyond the MVP artifact stage. The goal of this section is not to make unsupported promises, but to define a plausible research path for turning the current framework into a broader empirical program.

D.1 Axis 1: Model Diversity

One natural expansion axis is model diversity. The present artifact is designed to be portable across strong language models, but its behavior may vary depending on model family, instruction hierarchy, context handling, and structured-output reliability. Future benchmark design should therefore compare the runtime across multiple model classes rather than treating one model family as representative.

D.2 Axis 2: Task Diversity

A second axis is task diversity. The current MVP focuses on structurally contested diagnostic-style prompts. A broader benchmark could include:

- retrieval-grounded tasks
- code debugging tasks
- agent planning tasks

- policy or governance drafting tasks
- long-form explanation tasks with escalating ambiguity

This would help clarify where legality-first governance generalizes and where it remains narrow.

D.3 Axis 3: Human and Hybrid Evaluation

A third axis is evaluation-source diversity. The current evaluator is artifact-aligned and useful for internal legality assessment, but future work should combine:

- runtime self-report
- framework-aligned evaluator judgment
- human reviewer judgment
- downstream outcome observations

This would reduce dependence on a single evaluation lens.

D.4 Axis 4: Ablation of Runtime Components

A fourth axis is ablation. The framework becomes more scientifically informative if future experiments selectively remove or weaken components such as:

- neighboring-cut review
- repair-legality checks
- public-ceiling control
- long-context contamination guard
- structured output contract

Ablation would help clarify which parts of the runtime contribute most strongly to reduced illegitimate generation.

D.5 Axis 5: Dual-Layer Evaluation

A fifth axis is explicit benchmarking of the forward-atlas plus inverse-atlas pairing. Such experiments could compare:

- direct baseline
- forward-only guidance
- inverse-only governance
- forward-plus-inverse dual-layer operation

This would test the paper's claim that mapping and authorization are complementary rather than redundant.

D.6 Benchmark Philosophy Going Forward

Future benchmark expansion should preserve the same honesty discipline as the MVP paper. The aim should not be to force the framework into universal superiority claims, but to determine the precise zones in which legality-first governance changes model behavior in reliable and practically valuable ways. In that sense, benchmark expansion is not merely a scaling exercise. It is part of the framework's continued falsifiability.