GOV-Ω

Governance of the Ω Triple

Gate Minimality, Policy Invariance, and Audit-Grade Identifiability Under the AoR

Justin Grieshop

Marithmetics (public-arch/Marithmetics)

Abstract

The first question asked of any discrete-selection physics program is also the correct question: "How do we know you did not choose the rules because they produce the result you wanted?" This paper answers that question as a governance problem rather than as a narrative defense. We define an operational standard for selector governance—bounded degrees of freedom, gate minimality (shown by ablation collapse), policy invariance (equivalent outcomes under plausible tie-break policies), representation invariance (base-as-gauge), legality separation (admissible vs. illegal operators), and counterfactual "teeth." Each requirement is tied to a pinned Authority-of-Record (AoR) archive that supports independent audit. The central claim is narrow: within a declared family of constraints and policies, the Ω triple is *identifiable* rather than *chosen*.

`AoR commit: aor-20260209T040755Z | Bundle: GUM_BUNDLE_v30_20260209T040755Z`

Contents

Evidence Capsule (AoR + pinned artifacts; citation basis)

All citations in GOV-Ω refer to the entries below (`EC-*`). Each entry is commit-pinned and points to an auditable artifact, log, table, or script. All URLs below are rooted at:

https://github.com/public-arch/Marithmetics/tree/aor-20260209T040755Z

Identifier	Artifact
`EC-RepoRoot`	Repository root (pinned tag)
`EC-PublicationSpine`	publication_spine/
`EC-DOC`	Deterministic Operator Calculus specification (PDF)
`EC-AoRRoot`	Authority archive root
`EC-AoRMasterZip`	Master release archive (ZIP)
`EC-AoRReportPDF`	GUM Report v32 (PDF)
`EC-AoRReportManifest`	Report manifest (JSON)
`EC-ClaimLedger`	Claim ledger (JSONL)
`EC-RunMetadata`	Run metadata (JSON)
`EC-RunnerTranscript`	Runner transcript (TXT)
`EC-BundleSHAFile`	Bundle SHA-256 identity file
`EC-TableDemoIndexCSV`	tables/demo_index.csv
`EC-TableFalsificationMatrixCSV`	tables/falsification_matrix.csv
`EC-TableReproducibilityCSV`	tables/run_reproducibility.csv
`EC-D77-Script`	Demo-77 grammar rigidity script
`EC-D77-Stdout`	Demo-77 grammar rigidity stdout log
`EC-D78-Script`	Demo-78 midlift protocol script
`EC-D78-Stdout`	Demo-78 midlift protocol stdout log
`EC-D64-Stdout`	Demo-64 base-as-gauge invariance stdout
`EC-D69-Stdout`	Demo-69 OATB operator admissibility stdout
`EC-D34-Stdout`	Demo-34 $\Omega \to$ SM bridge stdout
`EC-D33-Stdout`	Demo-33 SM closure surface stdout

1 Reader Contract

1.1 The default objection is correct until it is operationally excluded

The most common failure mode for selector-based claims is post-hoc construction: rules are chosen because they yield a preferred output. When a reader asks, "How did you choose the rules?" they are not being cynical. They are applying a correct prior.

This paper does not attempt to reverse that prior by assertion. It reverses it—when possible—by converting it into a set of tests. If those tests fail, the correct conclusion is to stop.

1.2 What this paper is

GOV-Ω is a governance paper. It defines what it would mean for an Ω triple to be *identifiable* rather than *chosen*, under three constraints:

1. the selector family must be declared and bounded,
2. selection must be reproducible under a pinned audit record (AoR), and
3. legality must be enforced (admissible vs. illegal operators), so the selector is not effectively "tuned" by unconstrained transforms.

The supporting artifacts live in the AoR record (`EC-AoRRoot`) and the Deterministic Operator Calculus specification (`EC-DOC`).

1.3 What this paper is not

GOV-Ω is not a physics-results paper. It does not claim that any downstream bridge implies physical truth. It does not ask a reader to accept an interpretation. Its claims are narrower:

- what is fixed vs. what is allowed to vary,
- how gates behave under ablation,
- how selection behaves under policy variation,
- how selection behaves under representational changes, and
- how counterfactual perturbations degrade outcomes ("teeth").

1.4 Evidence discipline (governance is audit-first)

This paper treats governance as an audit problem. Evidence must be checkable without trusting narrative.

- **Pinned record:** AoR identity and replay surfaces are required (`EC-BundleSHAFile`, `EC-RunnerTranscript`, `EC-TableReproducibilityCSV`).
- **Declared surfaces:** the demo index and falsification matrix provide a map of what ran and what is required to fail (`EC-TableDemoIndexCSV`, `EC-TableFalsificationMatrixCSV`).
- **Primary governance runs:** Demo-77 (rigidity + enumeration + policy equivalence) and Demo-78 (midlift + illegals + teeth + promotion gating) provide the governance payload (`EC-D77-Stdout`, `EC-D78-Stdout`).

1.5 How to use GOV-Ω

A reader can treat this paper as a checklist:

1. Confirm that a unique Ω triple is produced in a declared primary window and that the run is pinned by deterministic hashes (`EC-D77-Stdout`).
2. Confirm that gate ablation causes predictable collapse of identifiability (`EC-D77-Stdout`).
3. Confirm that plausible policies converge rather than diverge (policy invariance) (`EC-D77-Stdout`).

4. Confirm that representation changes do not alter the selected triple (base-as-gauge) (EC-D64-Stdout).
5. Confirm that illegals are tested and rejected by legality, not by convenience (EC-D69-Stdout; EC-DOC).
6. Confirm that counterfactual perturbations degrade under the same budgets (teeth) (EC-D78-Stdout; EC-D77-Stdout).

Only after these checks does it become reasonable to treat the Ω triple as a governance-fixed input to downstream bridge work (EC-D34-Stdout).

2 Threat Model for Ω-Selection (turn the accusation into tests)

2.1 The accusation (stated plainly)

The governance accusation is:

The rules were selected because they produce the Ω triple (and/or downstream results).

If this were correct, certain patterns should appear reliably:

- ablations should not matter much,
- alternative policies should often pick different triples,
- representational changes should often change outcomes (if digits are being exploited), and
- counterfactual triples should frequently work nearly as well.

GOV-Ω makes these patterns explicit, then checks for their presence in the AoR record.

2.2 Threat model table (risk → required countermeasure → where to verify)

Governance risk (common critique)	**What you would expect if backsolved**	**Required countermeasure**	**Primary audit surfaces**
Gate arbitrariness ("you picked gates that force your answer")	Removing a gate does not meaningfully change candidate landscape	Ablation collapse: removing a gate increases candidate pools or destroys uniqueness in predictable ways	EC-D77-Stdout
Policy tuning ("you chose the tie-break policy that picks Ω")	Different plausible policies pick different winners	Policy invariance: plausible policies converge; disagreement is measurable and small	EC-D77-Stdout
Representation artifacts ("base-10 numerology")	Base change alters results; digit-based selectors can be hidden	Base-as-gauge invariance: outcomes survive base transport under declared tolerance	EC-D64-Stdout
Hidden smoothing knobs ("you tuned transforms")	Illegal transforms can improve naive metrics and are used implicitly	Legality separation: admissible operators pass; illegal controls fail for admissibility reasons	EC-D69-Stdout; EC-DOC

Governance risk (common critique)	What you would expect if backsolved	Required countermeasure	Primary audit surfaces
Soft selection (“many triples work similarly well”)	Counterfactual triples frequently survive budgets	Teeth: counterfactual perturbations degrade deterministically	EC-D78-Stdout; EC-D77-Stdout
Non-replayable selection (“trust me”)	Missing logs/hashes; unverifiable trace	AoR chain-of-custody: pinned identity, transcripts, reproducibility surfaces	EC-BundleSHAFile; EC-RunnerTranscript; EC-TableReproducibilityCSV

2.3 The governance standard (one paragraph)

An Ω triple is treated as governance-fixed only if it is: (i) identifiable within a declared family of gates and policies, (ii) stable under representational transport, (iii) brittle under corruption and counterfactual perturbation (teeth), and (iv) produced under a pinned, replayable audit record. The next sections define these terms and show how each requirement is audited in the AoR archive (EC-AoRRoot, EC-D77-Stdout, EC-D78-Stdout).

3 Governance Objects and Declared Rule Family

3.1 Governance objects (what is being governed)

This paper uses a small set of governance objects. Each object exists so that “how did you choose the rules?” becomes auditable rather than interpretive.

Ω triple. An ordered triple of lane representatives $(w_{U(1)}, w_{SU(2)}, w_{SU(3)})$ produced by an admissible selector under a declared window and gate set. In the AoR record used here, the baseline lock yields the unique ordered triple (137, 107, 103). (EC-D77-Stdout)

Lane pools. Candidate sets $S_{U(1)}$, $S_{SU(2)}$, $S_{SU(3)}$ produced by applying gates to the windowed candidate population for each lane. In the baseline lock, each lane pool collapses to a singleton [137], [107], [103]. (EC-D77-Stdout)

Gates. Explicit constraints applied to candidate generation and/or lane admissibility. Gates are not “preferences.” They are governance controls. Each gate must either (i) be shown load-bearing under at least one declared scenario or (ii) be marked redundant and treated accordingly. (EC-D77-Stdout)

Policies. Rules used only when more than one admissible triple survives (degeneracy breaking). Policies must be declared in advance and audited for equivalence, not tuned after the fact. (EC-D77-Stdout)

Admissible operators. Transform operators used in any bridge, smoothing, or transfer context. Admissibility is defined by DOC and enforced as non-negotiable. “Better-looking” illegal alternatives must be explicitly tested and rejected on legality grounds. (EC-DOC; EC-D69-Stdout)

3.2 The rule family is declared, bounded, and replayable

The governance accusation relies on the idea that the selector has many hidden knobs. The first governance requirement is therefore a degrees-of-freedom ledger.

This ledger is not a philosophy statement. It is a compact declaration of what may vary, what is forbidden, and where that variation is exercised in the AoR record.

Degrees-of-freedom ledger (declared flexibility vs. forbidden flexibility)

Channel	Declared variation (allowed)	Forbidden (non-auditable or circular)	Where audited
Window choice	Window shift and width within a declared scan family	Choosing a window because it yields a preferred triple or downstream outcome	Window scan ledger: 78 windows tested; 14 unique; all 14 canonical. (EC-D77-Stdout)
τ-grid thresholds	τ thresholds within a declared grid family	Picking a single "lucky" threshold post-hoc	Expanded τ scan: $21^3 = 9261$ cases; 5733 unique; 5733 canonical; 0 other unique. (EC-D77-Stdout)
Residue sets	Enumeration within explicit size bounds (e.g., $\lvert R\rvert \le 5$)	Hand-picking residues to force a target	Exact residue enumeration: 75,112,287,760 instances audited; uniqueness statistics recorded. (EC-D77-Stdout)
Gate presence	Ablation tests (drop one gate) are allowed as diagnostics	Quietly changing gates between runs without declaration	Multi-scenario gate classification + ablation outcomes recorded. (EC-D77-Stdout)
Policy choice	A small, declared set of plausible tie-break policies	Inventing a policy that exists only to pick a preferred triple	MinLane vs MinMDL equivalence: disagreement weight = 0. (EC-D77-Stdout)
Operator choice (bridges/ transfers)	Only DOC-admissible operators	Illegal filters / smoothing that improve a naive metric	Legality separation audits; illegal controls tested and rejected. (EC-DOC; EC-D69-Stdout)
External data overlays	Evidence-only diagnostics after selection	Any upstream influence on selection, budgets, or acceptance	Non-circularity contract; overlays are fenced. (EC-ClaimLedger; EC-AoRRoot)

The effect of declaring this ledger is simple: "how did you choose the rules?" becomes "which declared knobs exist, and what happens when we vary them?" Those outcomes are recorded in the AoR archive rather than argued in prose. (EC-AoRRoot; EC-D77-Stdout)

4 Gate Minimality (Necessity Shown by Ablation Collapse)

4.1 The governance principle: gates are justified by failure modes

In a selector program, a gate is governance-legitimate only if it prevents a specific, observable failure mode.

This paper therefore does not argue that gates are "reasonable." It shows that removing gates changes the survivor landscape in a predictable way, and that this change is recorded under explicit scenarios.

The central deliverable of this section is a **Gate Removal Impact Table** extracted from the multi-scenario gate classification in Demo-77. (EC-D77-Stdout)

4.2 Gate Removal Impact Table (multi-scenario; recorded lane pool sizes and triple counts)

Demo-77 reports three scenarios and the effect of dropping each gate. The quantities reported are the lane pool sizes ($|U1|$, $|SU2|$, $|SU3|$) and the resulting ordered triple count T. (EC-D77-Stdout)

Scenario S0 (baseline window): window [97, 181]
Baseline sizes: $U1$=1, $SU2$=1, $SU3$=1, T=1. (EC-D77-Stdout)

Ablation	Sizes $(U1, SU2, SU3, T)$	Governance interpretation
`DROP_C1_prime`	(1, 2, 2, 1)	Expands pools; triple still unique
`DROP_C2_residue`	(1, 6, 7, 3)	Multiple triples (loss of identifiability)
`DROP_C3_q_gt_sqrtw`	(1, 1, 1, 1)	Redundant in baseline
`DROP_C4_theta_floor`	(1, 1, 1, 1)	Redundant in baseline
`DROP_GvU1_...`	(3, 1, 1, 1)	Expands U(1) pool; still unique

Scenario S1 (extended window): window [50, 250]
Baseline sizes: $U1$=2, $SU2$=1, $SU3$=2, T=1. (EC-D77-Stdout)

Ablation	Sizes $(U1, SU2, SU3, T)$	Governance interpretation
`DROP_C1_prime`	(2, 3, 3, 1)	Pool expansion under window extension
`DROP_C2_residue`	(4, 11, 17, 96)	Candidate explosion; massive degeneracy
`DROP_C3_q_gt_sqrtw`	(2, 1, 2, 1)	Redundant in this scenario
`DROP_C4_theta_floor`	(2, 1, 2, 1)	Redundant in this scenario
`DROP_GvU1_...`	(5, 1, 2, 2)	Multiple triples (loss of identifiability)

Scenario S2 (shifted window): window [150, 234]
Baseline sizes: $U1$=0, $SU2$=0, $SU3$=0, T=0. (EC-D77-Stdout)

Ablation	Sizes $(U1, SU2, SU3, T)$	Governance interpretation
`DROP_C1_prime`	(0, 1, 1, 0)	Does not create triples in shifted region
`DROP_C2_residue`	(1, 2, 7, 1)	Allows triple where baseline has none
`DROP_C3_q_gt_sqrtw`	(0, 0, 0, 0)	Redundant here
`DROP_C4_theta_floor`	(0, 0, 0, 0)	Redundant here
`DROP_GvU1_...`	(0, 0, 0, 0)	Redundant here

4.3 Structural vs. redundant gates (classification recorded)

Demo-77 records a classification derived from the multi-scenario behavior above:

- **STRUCTURAL:** `DROP_C1_prime`, `DROP_C2_residue`, `DROP_GvU1_u1_v2_coherence`
- **REDUNDANT-BASELINE:** `DROP_C3_q_gt_sqrtw`, `DROP_C4_theta_floor`

(EC-D77-Stdout)

This distinction matters. A common failure mode in "defensive" governance is to treat all gates as equally sacred. Demo-77 does not do that. It explicitly identifies gates that are redundant in baseline and shows where other gates become load-bearing under realistic extensions (e.g., extended windows). That is the correct governance posture. (EC-D77-Stdout)

4.4 Designed-fail controls (gates are not only "passes")

A skeptic should not need to infer whether the authors tested "obvious cheats." The record should contain designed fails.

Demo-77 includes a negative controls suite:

- Baseline yields unique canonical triple.
- Lane swap, residue mirror, wrong U(1) coherence, and wrong window produce no canonical unique outcome.

(EC-D77-Stdout)

This is governance-relevant for one reason: it shows that the selection does not behave like a soft coincidence generator. If the pipeline is corrupted in simple, legible ways, it fails cleanly. (EC-D77-Stdout)

4.5 Interim governance conclusion (what this section establishes)

The multi-scenario gate ablation table establishes a minimal governance claim:

1. gates are explicit,
2. gate removal changes the survivor landscape in predictable ways, and
3. at least two gates (residue gating and U(1) coherence) are demonstrably load-bearing because their ablation produces degeneracy or explosion under declared scenarios.

(EC-D77-Stdout)

This is the first step in converting "you chose the rules" into an audit statement: the gates behave like structural constraints, not stylistic preferences.

5 Gate Necessity and Governance Minimality

The governance question is not "do you get impressive downstream numbers?" The governance question is: could any reasonably skeptical person reproduce the triple by following a fixed protocol, without giving themselves degrees of freedom to steer toward a desired answer?

This section makes the "necessity" claim precise:

- Each gate exists to block a specific failure mode that is common in numerology and in post-hoc computational discovery.
- The gate set is not justified by rhetoric. It is justified by counterfactual failure: when a gate is removed or corrupted, uniqueness collapses, or representation invariance collapses, or legality collapses.

The central posture is simple:

> *If a claim cannot survive a governance standard, it should be treated as numerology until proven otherwise.*

This is not defensive framing. It is the protocol that made the Ω-triple discovery possible in the first place.

5.1 Numerology failure modes and their corresponding gates

The table below is the "reader's checklist" version: the failure mode a skeptical physicist will assume, and the exact governance barrier that addresses it.

Failure mode ("red flag")	Looks like...	Governance barrier	Witness / where to audit
Hidden degrees of freedom	Windows, moduli, thresholds adjusted until something "pops"	Freeze points + scan ledger	Demo-77 scan stages + output locks (EC-D77-Stdout)
Soft matching	"Close enough" tolerances; vague similarity metrics	Exactness-first; explicit residual budgets and hard gates	DOC constraints + "teeth" (EC-DOC, EC-D77-Stdout, EC-D78-Stdout)
Illicit operators	Smoothing, fitting, or transforms that silently inject bias	Operator legality: admissible kernels only; illegal suite required and must fail	Demo-78 illegal suite + breaks (EC-D78-Stdout)
Post-hoc rule selection	Rules chosen because they output the desired values	Policy equivalence: independent policies must agree	Demo-77 P_MinLane vs P_MinMDL (EC-D77-Stdout)
Representation dependence	"Works in base 10" / fails under other representations	Base-as-gauge requirement	Demo-64; cited there, enforced here (EC-D64-Stdout; EC-TableFalsificationMatrixCSV)
External data leakage	CODATA/Planck numbers implicitly influence "selection"	Non-circularity contract: external data is overlay-only	AoR claim ledger + falsification matrix (EC-ClaimLedger, EC-TableFalsificationMatrixCSV)
Un-auditable computation	"Trust me" outputs without a stable record	Authority-of-Record anchoring	AoR bundle + logs + metadata (EC-AoRRoot, EC-BundleSHAFile, EC-RunMetadata, EC-RunnerTranscript)

This is the core psychological move the paper must make: we do not ask the reader to accept an extraordinary claim; we give them a protocol designed to fail loudly when someone tries to cheat.

5.2 What "minimality" means (and what it does not mean)

When we say "the gates are necessary", we do not mean that every gate is metaphysically fundamental. We mean:

1. **Necessary for uniqueness under audit.** If you remove the gate, the candidate space expands in a way that makes selection non-unique or manipulable.
2. **Necessary for invariance under representation.** If you remove the gate, the result becomes representation-dependent (base, encoding, coordinate choice).
3. **Necessary for legality under DOC constraints.** If you remove the gate, you allow operators that violate the admissibility contract (e.g., non-PSD kernels, sign corruption).

Those are governance statements. They are not "physics is true because this is pretty."

Demo-77 and Demo-78 are included in this governance section specifically because they supply the most direct, least interpretive kind of evidence: remove/alter a constraint → the property collapses. (EC-D77-Stdout, EC-D78-Stdout)

5.3 Policy equivalence as the anti-"we picked the rules" theorem (practical form)

The single most important governance device in this paper is **policy equivalence**:

- We define multiple selection policies that are not identical in motivation (e.g., one is "lane minimization," another is "description length minimization").
- We require them to agree on the canonical selection—or else we quantify disagreement and treat it as a governance failure.

Demo-77 implements this explicitly in the degeneracy-break audit:

- It reports a large `decided_count` (cases where a decision is required rather than trivial).
- It then reports `policy_disagreement_weight = 0`, i.e., the policies agree on the decisions that matter.
- It separately reports canonical hit counts under each policy, including the shared canonical outcome (137, 107, 103).

(EC-D77-Stdout)

This is what "answering the first question definitively" looks like: not a philosophical defense of how rules were chosen, but a demonstrable fact that distinct governance policies collapse onto the same canonical outcome in the audit regime.

6 DEMO-78 as a Second, Orthogonal Governance Witness: the Midlift Protocol

Demo-77 shows that the Ω-triple is not a casual artifact of flexible selection. Demo-78 is included for a different reason: it shows that operator legality and "teeth" constraints are not cosmetic—they materially constrain the pipeline.

6.1 What "midlift" means here

In this program's governance vocabulary, "midlift" refers to the first audited scale transition between a strict finite substrate representation and a representation that is usable for downstream physical mappings.

It is "first" in the operational sense: it is the earliest point where one can accidentally (or deliberately) insert smoothing, reweighting, or other choices that create hidden degrees of freedom. This is why Demo-78 is governance-relevant.

We do not claim "the first physical dimension is midlift" as a standalone metaphysical statement in this governance paper. We claim the narrower, audit-grade statement:

- the midlift layer is the first place where legality must be enforced aggressively,
- because it is the first place where illegal operators can superficially improve metrics while breaking the DOC admissibility contract.

6.2 Compression equivalence: invariance under representation change inside the pipeline

Demo-78 begins with an explicit compression equivalence test:

- `rel_frob` $\approx 6.6897 \times 10^{-17}$
- `GATE C0: True`

(EC-D78-Stdout)

This is a governance-relevant statement: it shows that an internal representation change (compression) does not create a measurable drift. That is exactly the type of invariance that blocks "we encoded it differently until it worked."

6.3 Illegal operator suite: "show the fail" where cheating would be easiest

Demo-78 then runs an explicit multi-illegal suite and reports whether each illegal choice "breaks" the admissibility properties:

- `SIGNED`: break=True with `min_eig` negative
- `NEG_DIAG`: break=True with `min_eig` negative
- `CHECKER`: break=False (a useful control: not every perturbation is catastrophic)
- `GATE I0: True` (meaning the suite behaved as expected)

(EC-D78-Stdout)

This is not window dressing. A governance paper needs to force the reader to see that the program is not designed to preserve success at all costs; it is designed to preserve legality and report failure.

6.4 Φ-map: primary legal mapping vs illegal mapping degradation

Demo-78 reports a Φ-map under the primary legal pipeline and under an illegal signed variant.

Primary legal Φ-map (all `ok=True`):

- $\alpha_{\rm em}$ mid ≈ 0.00734658, err $\approx 4.73 \times 10^{-5}$
- $\sin^2\theta_W$ mid ≈ 0.23340485, err $\approx 7.15 \times 10^{-5}$
- α_s mid ≈ 0.11845213, err $\approx 8.05 \times 10^{-4}$
- `GATE M0Φ: True`

(EC-D78-Stdout)

Illegal signed Φ-map degrades dramatically:

- $\alpha_{\rm em}$ err ≈ 0.0358
- α_s err ≈ 0.218
- `GATE M1Φ: True`

(EC-D78-Stdout)

Governance point

The legality constraint differentiates between pipelines in a way that cannot be faked by "just smoothing harder."

This is a direct rebuttal to the most common computational-physics suspicion: "you tuned your kernel."

6.5 Counterfactual teeth: discrete sensitivity as an anti-coincidence witness

Finally, Demo-78 includes explicit "teeth" counterfactuals—nearby triples that should fail if the selection is rigid rather than fuzzy:

- CF (409, 263, 239): miss=True
- CF (409, 263, 307): miss=True
- CF (409, 367, 239): miss=True
- CF (409, 367, 307): miss=True
- `TEETH Φ: 4/4; GATE T0Φ: True`

(EC-D78-Stdout)

This is one of the strongest "pause payloads" available in governance form, because it shows a property that is hard to reconcile with vague pattern matching:

- A rigid selection rule exhibits cliffs.
- A numerology rule exhibits smoothness (you can usually perturb inputs without catastrophic loss because the "fit" was never tied to invariants).

Teeth is therefore not a rhetorical flourish. It is a governance device: it creates a visible falsification boundary.

6.6 Self-contained ledger freezing: preventing hidden target drift

Demo-78 also reports that the core physical map is based on a self-contained ledger with a recorded sha256:

- `core_ledger_sha256` reported
- `core_mined_specs_found=6/6`
- frozen specs sha256 recorded

(`EC-D78-Stdout`)

This is governance-critical. Without it, a critic can always say: "you moved the targets."

Ledger freezing makes that accusation auditable: either the ledger was fixed before the run, or it was not.

7 What the Governance Evidence Actually Establishes (and What It Leaves to Physics Papers)

This paper is not PH-2 (constants), not PH-3 (cosmology), not PH-4 (observer/controller legality), and not DOC (operator calculus). It is the governance spine that answers the first question a serious reader asks:

> *"How did you choose the rules?"*

The governance answer, in audit-grade form, is:

1. The gates are specified before results are evaluated as "success."
2. The gates are justified by counterfactual collapse when removed or corrupted.
3. Independent policies converge (policy equivalence), collapsing the accusation of post-hoc rule choice.
4. Illegal operators are explicitly tested and shown to break, preventing the "kernel tuning" escape hatch.
5. The ledger and artifacts are frozen, hashed, and anchored to AoR, preventing silent drift.

Demo-77 supplies (2) and (3) at large scale (enumeration + policy equivalence + scan ledger). (`EC-D77-Stdout`)

Demo-78 supplies (4) and (5) at the first scale transition where illegal tuning would be easiest. (`EC-D78-Stdout`)

8 Integration Boundary: Ω is Upstream and Non-Intervening

The governance accusation can reappear in a subtler form:

> *Even if the Ω triple looks "identifiable," you might still be using downstream physics success to justify upstream choices.*

This section addresses that concern as an integration boundary. The boundary is a governance rule:

Boundary rule

Downstream bridges may consume the Ω triple, but they may not select it.

8.1 The boundary rule (what is allowed and what is forbidden)

Allowed (governance-safe):

- A downstream bridge takes the Ω triple as a fixed input.
- The bridge is evaluated under admissible operators and pinned artifacts.
- The bridge may be inspected as a consequence of the governance-fixed input.

Forbidden (governance-breaking):

- Using downstream metrics to choose gates, policies, windows, or thresholds upstream.
- Selecting a policy because it yields a "better bridge."
- Editing the selector family after inspecting downstream overlays.

This boundary is enforced operationally by two requirements:

1. selection-relevant artifacts are pinned in the AoR, and
2. the falsification matrix records illegality classes and disallowed selection pathways.

(`EC-AoRRoot`; `EC-ClaimLedger`; `EC-TableFalsificationMatrixCSV`)

8.2 A practical integration witness: Ω → SM bridge consumes Ω but does not select it

The most useful integration witness is not a long physics argument. It is a short trace showing that the Ω triple is treated as an upstream certificate and is then consumed by a downstream computation.

Demo-34 provides that role in this governance stack: it is a bridge snapshot surface that demonstrates downstream consumption of Ω under the same audit posture used elsewhere. (`EC-D34-Stdout`)

For readers who want the deeper Standard Model closure surface, Demo-33 exists as a pinned audit object. GOV-Ω does not require that deeper read for the governance claim, but it is included so the integration boundary cannot be dismissed as "hand-waving." (`EC-D33-Stdout`)

8.3 Why this matters

If the boundary is not explicit, a critic can always claim hidden circularity:

- "You chose the gates because the bridge looked good."
- "You chose the triple because it supports downstream closure."

The boundary makes these claims auditable. A reader may still disagree with downstream interpretation, but the governance question becomes separate:

- Is Ω identifiable under the selector governance standard?
- Does the AoR record show that identifiability without referencing external data?

Those questions are answered upstream by Demo-77 and Demo-78, and their associated governance surfaces. (`EC-D77-Stdout`; `EC-D78-Stdout`)

9 Hostile Referee Protocol (Selector Governance Audit)

This section is designed to be executable. It is intentionally procedural. The goal is not to persuade; it is to let a skeptical reader reproduce the governance verdict.

9.1 Pre-flight (2 minutes): confirm the audit object exists

1. Confirm the repository tag is pinned: (`EC-RepoRoot`).

2. Confirm the AoR root exists and matches the paper's AoR commit: (`EC-AoRRoot`).
3. Confirm the bundle identity file exists: (`EC-BundleSHAFile`).
4. Confirm the run is replay-oriented: run transcript and reproducibility table exist: (`EC-RunnerTranscript`; `EC-TableReproducibilityCSV`).
5. Confirm the falsification matrix exists (governance requires "must fail" surfaces): (`EC-TableFalsificationMatrixCSV`).
6. Confirm the governance demos exist in the demo index (avoid filename inference): (`EC-TableDemoIndexCSV`).

If any pre-flight step fails, stop. The governance object is not defined.

9.2 The 20-minute governance audit (minimum viable verdict)

Step 1 — Baseline lock (uniqueness under a declared window).
Open Demo-77 stdout. Confirm a declared primary window exists and yields singleton lane pools and a unique ordered triple. In the AoR record used here, the baseline lock yields (137, 107, 103). (`EC-D77-Stdout`)

Step 2 — Gate ablation collapse (load-bearing constraints).
In Demo-77, locate the multi-scenario gate classification. Confirm at least one gate removal causes degeneracy explosion or loss of identifiability under declared scenarios. The residue gate and U(1) coherence gate are recorded as structurally load-bearing via ablation outcomes. (`EC-D77-Stdout`)

Step 3 — Policy invariance (tie-break policies converge).
In Demo-77, locate the degeneracy-break audit. Confirm that plausible policy alternatives agree (or that disagreement is quantified and small). In the AoR record, MinLane vs MinMDL disagreement weight is recorded as zero. (`EC-D77-Stdout`)

Step 4 — Counterfactual teeth (discrete sensitivity).
Open Demo-78 stdout. Confirm that counterfactual perturbations degrade under the declared teeth test, rather than sometimes improving randomly. In the AoR record, the Φ teeth stress returns 4/4 misses for counterfactuals and passes the corresponding gate. (`EC-D78-Stdout`)

This 20-minute audit is sufficient to decide whether the selector behaves like an identifiable mechanism or like a post-hoc coincidence generator.

9.3 The 45-minute governance audit (recommended)

Perform the minimum audit, then add:

Step 5 — Representation invariance (base-as-gauge).
Open the base-as-gauge invariance run. Confirm invariance passes under representational transport within declared tolerance. (`EC-D64-Stdout`)

Step 6 — Legality separation (admissible vs. illegal operators).
Read the operator admissibility / transfer audit. Confirm that illegals are explicitly tested and rejected for admissibility reasons rather than "fit quality." Use DOC as the admissibility specification. (`EC-D69-Stdout`; `EC-DOC`)

This step closes the "digit/base trick" and "kernel tuning" escape hatches at the governance level.

9.4 The 60-minute integration boundary check (optional but clarifying)

If a reader wants to verify that Ω is not being selected by downstream bridge outcomes:

Step 7 — Downstream consumption without selection.
Open the Ω → SM bridge snapshot and confirm it behaves as a downstream consumer surface. (`EC-D34-Stdout`) Optionally inspect the deeper SM closure surface as a pinned continuation. (`EC-D33-Stdout`)

This step is not required to establish selector identifiability, but it is useful to rule out the "downstream leakage" narrative.

9.5 What counts as a governance failure

The governance audit fails if any of the following occur:

- No declared baseline lock exists, or uniqueness is not demonstrated. (`EC-D77-Stdout`)
- Gate ablation does not materially change identifiability (suggesting gates are not load-bearing). (`EC-D77-Stdout`)
- Different plausible policies select different winners without quantified disagreement (suggesting policy tuning). (`EC-D77-Stdout`)
- Counterfactuals frequently succeed under the same budgets (no teeth). (`EC-D78-Stdout`)
- Representation transport changes the selection (representation dependence). (`EC-D64-Stdout`)
- Illegal operators are not explicitly tested and disqualified by admissibility (hidden smoothing degrees of freedom). (`EC-D69-Stdout`; `EC-DOC`)
- AoR identity and replay surfaces are missing (non-auditable selection). (`EC-BundleSHAFile`; `EC-RunnerTranscript`; `EC-TableReproducibilityCSV`)

A governance failure does not imply "the idea is wrong." It implies the correct posture is to treat the selector as non-identifiable and therefore treat any downstream narrative as ungrounded until the governance standard is met.

10 Additional Governance Exhibits

This section is brief by design. The core governance argument is already established by (i) gate ablation collapse, (ii) policy equivalence, and (iii) counterfactual teeth. This section closes the two remaining escape hatches that commonly sustain the "you chose the rules" narrative in practice:

- representation dependence ("base-10 numerology"), and
- hidden smoothing or transform degrees of freedom ("you tuned the kernel").

10.1 Representation invariance (base-as-gauge)

Governance claim: if selection depends on representation, it is not a governed mechanism; it is a formatting artifact.

Required behavior: outcomes must survive base transport under declared tolerance. This is enforced as a first-class audit surface, not as a philosophical preference. (`EC-D64-Stdout`)

What to verify (minimal):

1. the demo explicitly performs representational base transport,
2. the selector output remains invariant under that transport, and
3. digit-anchored or injection-style perturbations are treated as illegality classes and tracked as required failures in the falsification matrix.

(`EC-D64-Stdout`; `EC-TableFalsificationMatrixCSV`)

This is a governance condition because it prevents the most common backsolving pathway in integer-based claims: encoding external targets into a representation choice.

10.2 Operator legality boundary (DOC + OATB)

Governance claim: legality must be enforced where "cheating transforms" are easiest to hide. In practice, this means that smoothing kernels, transfer operators, and bridge transforms cannot be treated as free knobs.

DOC provides the admissibility specification. The OATB audit provides the enforcement surface. (EC-DOC; EC-D69-Stdout)

Required behavior:

- admissible operators pass,
- illegal controls fail because they violate admissibility, and
- illegals are included even when they could superficially improve a naive metric.

What to verify (minimal):

1. DOC defines the admissibility constraints used in transfer and bridge contexts (EC-DOC),
2. Demo-69 explicitly tests illegal alternatives (controls) and disqualifies them by legality witnesses rather than by convenience (EC-D69-Stdout), and
3. the illegality classes are traceable in the falsification matrix and are not "hand-waved away." (EC-TableFalsificationMatrixCSV)

This closes the "you tuned the filter until it worked" narrative at the governance layer, where it must be closed before downstream bridges are evaluated.

11 Role in the Governance Stack (GUM + DOC + GOV-Ω)

This paper is intended to sit alongside GUM and DOC in the governance section:

- **GUM** (AoR governance): defines the Authority-of-Record packaging, the claim ledger, the manifest discipline, and the reproducibility surface that makes audit possible. (EC-AoRReportPDF; EC-ClaimLedger; EC-TableReproducibilityCSV)
- **DOC** (legality governance): defines admissible operators and the non-negotiable legality boundary used in transfer and bridge contexts. (EC-DOC)
- **GOV-Ω** (selector governance): establishes that the Ω triple is identifiable—i.e., fixed upstream by audited constraints and policies—rather than selected post-hoc by steering. (EC-D77-Stdout; EC-D78-Stdout)

This paper is also designed to be consistent with the Physics-Track audit contract (PH-0).

12 Scope, Claims, and Non-Claims

12.1 What GOV-Ω claims (and what would falsify it)

GOV-Ω makes a small set of governance claims, each tied to pinned audit surfaces:

1. **Baseline identifiability:** under a declared primary window and gate set, lane pools collapse to a unique ordered triple; this is recorded with deterministic certificate posture. (EC-D77-Stdout)
2. **Gate minimality by ablation collapse:** dropping structural gates produces predictable loss of identifiability (degeneracy or candidate explosion) under declared scenarios; redundant gates are explicitly identified as such. (EC-D77-Stdout)
3. **Policy invariance:** distinct plausible degeneracy-break policies converge (disagreement weight is measured and recorded). (EC-D77-Stdout)
4. **Counterfactual teeth:** near-neighbor or counterfactual triples degrade under the same budgets, rather than frequently succeeding in a soft-matching way. (EC-D78-Stdout)
5. **Legality separation:** admissible operators are enforced as a governance boundary; illegal controls are tested and rejected on admissibility grounds. (EC-DOC; EC-D69-Stdout)
6. **Audit-grade chain-of-custody:** the selection and its diagnostics live inside a pinned AoR record that supports independent inspection. (EC-BundleSHAFile; EC-RunnerTranscript; EC-RunMetadata; EC-AoRRoot)

What would falsify these claims:

- a comparable AoR replay in which a minor, declared governance variation (policy swap, window shift within declared scan family, or other declared knob) reliably selects a different

triple without corresponding collapse behavior, or
- a demonstration that the observed uniqueness is not stable under representation transport, or
- a demonstration that illegal operators can be substituted without violating admissibility witnesses and without being detected as illegality.

The purpose of making falsification explicit is governance hygiene: a governance standard that cannot be falsified is not governance.

12.2 What GOV-Ω does not claim

GOV-Ω does not claim:
- that the Ω triple is physically "true" by virtue of being identifiable,
- that downstream bridges validate upstream selection,
- that all possible policies would converge (only the declared policy set is audited here),
- that the selector family is the only possible family worth investigating,
- that continuity or continuum formalisms are invalid, or
- that agreement with any external dataset is required for governance identifiability.

GOV-Ω is intentionally narrower: it establishes whether the Ω triple is fixed by the selector governance standard under a pinned audit record.

13 Data and Materials Availability

All artifacts cited in GOV-Ω are available in the pinned repository and the AoR archive:
- Repository root (pinned tag): (`EC-RepoRoot`)
- AoR archive root: (`EC-AoRRoot`)
- Offline master archive (zip): (`EC-AoRMasterZip`)
- AoR report + manifest: (`EC-AoRReportPDF`; `EC-AoRReportManifest`)
- Deterministic identity surface: (`EC-BundleSHAFile`)
- Run context and replay surfaces: (`EC-RunMetadata`; `EC-RunnerTranscript`; `EC-TableReproducibilityCSV`)

Primary governance witnesses:
- Demo-77 (gate classification, ablation collapse, policy equivalence, scan ledgers, negative controls): (`EC-D77-Stdout`; `EC-D77-Script`)
- Demo-78 (midlift governance, illegal suite, ledger freezing, teeth): (`EC-D78-Stdout`; `EC-D78-Script`)
- Base-as-gauge invariance surface: (`EC-D64-Stdout`)
- Operator admissibility / transfer (OATB) surface + DOC specification: (`EC-D69-Stdout`; `EC-DOC`)

A Notation and Citation Discipline

A.1 Evidence Capsule identifiers

GOV-Ω cites primary artifacts using Evidence Capsule identifiers (`EC-*`). Each EC entry is a commit-pinned URL to a log, table, script, or AoR artifact.

When citing a behavior:
- cite a demo stdout log for a governance outcome (e.g., ablation collapse): (`EC-D77-Stdout`)
- cite a script only to support the declared procedure (e.g., how a scan was enumerated): (`EC-D77-Script`; `EC-D78-Script`)
- cite DOC for admissibility definitions: (`EC-DOC`)

- cite the falsification matrix for illegality classes and required failures: (EC-TableFalsificationMatrixCSV)

A.2 Falsification matrix reference

Throughout this paper, "the falsification matrix" refers to: `tables/falsification_matrix.csv`: (EC-TableFalsificationMatrixCSV)

B Verification Shortcuts (practical)

1. **Start with Demo-77.** Confirm baseline identifiability, then read the gate classification and policy equivalence blocks. This resolves the "how did you choose the rules?" question fastest. (EC-D77-Stdout)
2. **Then read Demo-78.** Confirm illegal suite behavior, teeth, and ledger freezing. This closes the "you tuned the kernel" and "soft matching" narratives. (EC-D78-Stdout)
3. **Then close the representation loophole.** Confirm base-as-gauge invariance. (EC-D64-Stdout)
4. **Then confirm legality is explicit.** Use DOC to understand admissibility, and Demo-69 to confirm enforcement. (EC-DOC; EC-D69-Stdout)

Conclusion

Selector-based claims are easy to backsolve. A serious reader is correct to assume rule choice was steered until an audit standard is met. GOV-Ω defines that standard for Ω-triple selection and ties each requirement to a pinned AoR record.

Under the AoR commit audited here, the governance surfaces exhibit the behaviors that post-hoc selection typically cannot sustain under hostile inspection:

- uniqueness under a declared baseline lock,
- predictable collapse of identifiability under gate ablation,
- convergence under distinct plausible degeneracy-break policies,
- explicit illegality suites and legality enforcement,
- counterfactual teeth under the same budgets, and
- chain-of-custody anchoring suitable for independent audit.

If any of these governance requirements fail under replay, the correct conclusion is to treat downstream narratives as ungrounded until governance is repaired. If they hold, then the "you chose the rules because you wanted this triple" objection is no longer the default rational posture; it becomes a claim that must contradict audited, pinned evidence. (EC-D77-Stdout; EC-D78-Stdout; EC-AoRRoot)