Compitum: A Lyapunov-Stable Bounded Observer for Verifiable Routing

A self-published artifact on routing whose behavior is *proved as runnable tests*, not asserted

Paul Carver Tiffany III
Independent Researcher
compitum.space | github.com/PaulTiffany/compitum

June 7, 2026

 | Part of the Compitum research program | *Incomplete by construction*

Abstract

We describe *Compitum*, a geometrically-aware router for large language model (LLM) requests, and make an unusual claim about it: not that it wins a benchmark, but that its operational behavior is *falsifiable by construction*. Compitum selects a model by minimizing a scalarized *Symbolic Free Energy* F_s over a learned symmetric-positive-definite (SPD) metric, subject to feasibility-first constraints, while a Lyapunov-style controller gates two-timescale online updates. We state the assumptions under which F_s together with the controller's energy acts as a Lyapunov functional for the discrete routing dynamics, and—this is the point—we pin every claim to a property-based test that a reader can run to try to break it. We report an honest empirical picture: on a synthetic benchmark Compitum attains higher utility-per-dollar at zero constraint violations *at a measurable latency cost*; and on flat, engineered-feature materials data the geometric advantage *does not* appear, which we argue is a prediction of the framework rather than a failure of it. The artifact is offered to the noosphere in the spirit of *Come*: there is no perfect paper, only a substrate a reader can mutate and verify.

"All of us is limited. None of us has all the answers. Come develop with us."

1 Introduction

The dominant way to argue that a router is good is to report a number on a leaderboard. This invites a quiet form of dishonesty. A single aggregate beats hide the conditions under which a method helps or hurts; error bars are easy to draw and hard to earn; and a reader cannot, from the paper alone, check whether the claimed behavior actually holds in the code. We have come to call this pattern *masking*: asserting an adequacy the artifact cannot back.

This paper takes the opposite stance, which we call *verifiable by construction*. Compitum's contribution is not a benchmark win—we will be explicit about where it does *not* win—but a router whose structural and dynamical properties are stated as assumptions, derived as lemmas, and then *pinned to executable property-based tests*. The predictions in Section 4 each carry an explicit falsification criterion and the path to the test that checks it. If the claims are false, the reader can find out in minutes, without trusting us.

Request → **SPD chart** → **Free energy** → **Constraints** → **Certificate**
$x \in \mathbb{R}^d$ → $M = LL^\top + \delta I$ → $\arg\min_m F_s(x, m)$ → feasible set first → route + tests

Geometry	Control	Falsification	Scope
anisotropic model fit	line search + trust radius	P1–P8 invariant suite	local; surrogate; open

Figure 1: **Compitum at a glance.** The router acts as a bounded observer: it sees a finite request embedding, scores model fit in a learned SPD chart, minimizes a scalarized energy only over feasible actions, and emits diagnostics whose structural claims are pinned to tests.

Bounded observers. Compitum belongs to a wider program that treats knowers—electrons, networks, institutions, routers—as *bounded observers*: systems that must act on a coarse, finite read of a richer reality. A router is a bounded observer over a model pool: it never sees true quality, only calibrated estimates; it cannot try every model, only score a few; it must respect hard constraints (region, policy, capability) it did not choose. The companion artifacts *The Wicked Prior* [3] and the SRMF/geometric-systems notes develop the bounded-observer geometry in narrative and physical settings; the present artifact develops it for routing.

What we claim, and what we do not. We claim: (i) a concrete geometric construction (SPD metric learning + constraint-aware selection + a Lyapunov-style controller); (ii) lemmas, with stated assumptions, under which a composite energy is non-increasing along the routed trajectory; (iii) a falsification harness mapping each property to a runnable test; and (iv) an honest empirical reading, including a negative result. We do *not* claim a general benchmark victory, a complete (global) stability proof, or that F_s is the "correct" potential. Those are stated as open, in Section 6, as the direction the program is heading.

2 Construction

State and observers. A request is summarized by an embedding $x \in \mathbb{R}^d$. A model pool $\mathcal{A} = \{m_1, \ldots, m_{|\mathcal{A}|}\}$ is the action set. For each model m we hold a center $\mu_m \in \mathbb{R}^d$ and calibrated predictors $\hat{q}_m, \hat{\ell}_m, \hat{c}_m$ for quality, latency, and cost (`src/compitum/predictors.py`).

Learned SPD metric. Each model carries a low-rank SPD metric

$$M_m = L_m L_m^\top + \delta I, \qquad \delta > 0, \tag{1}$$

so M_m is strictly positive definite (`src/compitum/metric.py`). Distances are Mahalanobis on the whitened residual $z = x - \mu_m$: $d_{M_m}(x, \mu_m) = \sqrt{z^\top M_m z}$. We work with a learned SPD (Riemannian) metric rather than a fixed Euclidean one because the relevant notion of "fit" between a prompt and a model is anisotropic; the geometry is the inductive bias.

Symbolic Free Energy. Routing scalarizes competing desiderata into a single potential (`src/compitum/energy.py`):

$$F_s(x, m) = \alpha\,(-\hat{q}_m) + \beta_t\,\hat{\ell}_m + \beta_c\,\hat{c}_m + \beta_d\,d_{M_m}(x, \mu_m) - \beta_s\,e_m, \tag{2}$$

with nonnegative weights, where e_m is an evidence/practice term. Lower F_s is better: high predicted quality and evidence pull it down; latency, cost, and geometric distance push it up. The selected model is the feasible minimizer of F_s.

Feasibility-first selection. Hard constraints (region, policy, capability) are enforced before optimization by a reflective constraint solver (`src/compitum/constraints.py`), which returns the feasible set and approximate shadow prices (duals) [2]. Selection is therefore *constraint-aware*: an infeasible model is never chosen regardless of its F_s, and the duals are exposed as diagnostics rather than relied upon as exact.

Lyapunov-style controller and two timescales. Online adaptation runs on two timescales (`src/compitum/control.py`, `src/compitum/router.py`):

- **Fast (every step):** a controller maintains a drift integral with anti-windup and a trust radius $r \in [0.2, 5.0]$, shrinking r on violation ($\times 0.5$) and expanding on success ($\times 1.05$). Its Lyapunov candidate is $V_{\text{ctrl}} = (\text{drift integral})^2$.
- **Slow (every update_stride steps):** the metric factor L_m is updated by a backtracking line search on a whitened surrogate energy $E(L; z) = \frac{1}{2}\beta_d \|L^\top z\|^2$, accepting only steps with $E_{k+1} \leq E_k$.

Separating the timescales lets the controller damp drift between metric updates, and lets metric updates be conservative (line-searched) rather than aggressive.

3 Lyapunov-Style Stability

We state the discrete-time stability picture at the altitude the code actually supports: *local, surrogate, and matched by tests*, in the standard discrete Lyapunov/LaSalle sense [1]. Full continuous-time grounding is deferred to Section 6 as program direction, not present claim.

Assumptions. (A1) *Positive-definiteness*: $M_m = L_m L_m^\top + \delta I \succ 0$ for $\delta > 0$. (A2) *Observability*: F_s and per-action deltas are computable from calibrated predictors and the metric. (A3) *Bounded controller*: the trust radius is clipped to $[0.2, 5.0]$ and the drift integral is anti-windup limited. (A4) *Fixed selection between strides*: along the analyzed segment the chosen model and embedding are held fixed (no switching), isolating the learning/control dynamics.

Lemma 1 (Learning descent)**.** *Under (A1)–(A2) and a backtracking line search that accepts only non-increasing steps, the metric update satisfies* $E(L_{k+1}; z) \leq E(L_k; z)$.

Pinned to:
`tests/invariants/test_invariants_lg.py::test_metric_update_line_search_non_increase`

Lemma 2 (Zero-error Lyapunov decay)**.** *Under (A3), if the observed error proxy* $d_\star$ *is zero for* T *steps, the controller candidate* $V_{\text{ctrl}} = (\textit{drift integral})^2$ *is non-increasing over those steps.*

Pinned to:
`tests/invariants/test_invariants_control_sy.py::test_lyapunov_decays_under_zero_error`

Lemma 3 (Two-timescale isolation)**.** *Before* ***update_stride*** *is reached,* L_m *is unchanged; updates occur only at stride boundaries, subject to Lemma 1.*

Pinned to:
`tests/invariants/test_invariants_control_sy.py::test_two_timescale_metric_update_stride`

Proposition 4 (Composite non-increase)**.** *Define* $V_{\text{total}}(k) = E(L_k; z) + \gamma\, V_{\text{ctrl}}(k)$ *with* $\gamma > 0$. *Under (A1)–(A4), at the sequence of stride boundaries* V_{total} *is non-increasing and bounded below by* 0*: the controller term does not increase under small drift (Lemma 2) and the learning term does not increase at strides (Lemma 1). In the routed certificate, the negative of the distance component is an observable proxy for* E*, and it does not increase over repeated updates.*

Pinned to:
`tests/invariants/test_invariants_srmf_lyapunov.py::test_distance_decreases_over_updates`

Remark (Honest scope)*.* This is a *Lyapunov-style* statement: it establishes monotone non-increase of a composite surrogate under bounded-observer assumptions, matched by tests. It is local (a forward-invariant sublevel neighborhood), it assumes fixed selection between strides, and it does not by itself prove global asymptotic convergence. We say "F_s acts as a Lyapunov functional" in exactly this bounded sense, and no stronger.

4 Falsification Harness

The invariants suite is the empirical heart of this artifact: each property below is a prediction with an explicit way to be wrong, and a test that tries to make it wrong (most are property-based, generating many cases per run). To attempt falsification:

```
HYPOTHESIS_PROFILE=ci pytest -q tests/invariants
```

	Prediction	**Falsified if**	**Test**
P1	The learned metric M_m is strictly SPD.	any eigenvalue ≤ 0 for $\delta > 0$.	`test_invariants_metric`
P2	d_{M_m} obeys symmetry, triangle inequality, and ray monotonicity.	any axiom violated beyond tolerance.	`metric_triangle`, `metric_ray`
P3	(Lemma 1) Metric line-search updates are non-increasing in E.	$E_{k+1} > E_k$ on any accepted step.	`test_invariants_lg`
P4	(Lemma 2) V_{ctrl} decays under zero drift.	V_{ctrl} increases under $d_\star = 0$.	`control_sy`
P5	(Lemma 3) Learning is isolated between strides.	L_m changes before `update_stride`.	`control_sy`
P6	(Prop. 4) Routed distance proxy does not increase over updates.	final > initial beyond tolerance.	`srmf_lyapunov`
P7	Selection is feasibility-monotone; duals are near-zero slack at the boundary.	an infeasible model selected, or slack far from 0.	`constraints`, `duals_near_binding`
P8	Routing is deterministic and paraphrase behavior is bounded and explainable.	identical input yields different routes; flip budget exceeded.	`router_determinism`, `paraphrase_*`

A passing suite does not prove the framework "correct"; it proves these specific, named properties hold across the generated cases. That is a smaller and more honest claim than a benchmark number, and a more useful one: it tells you exactly what you can rely on.

Synthetic case A: utility-per-dollar at zero violations

Strategy	**UP$**	**visual scale**
Compitum	0.96	
Fixed best	0.82	
Greedy	0.82	
UCB1	0.84	
Thompson	0.82	

Tradeoff: Compitum's p95 latency is about 6.1 s in this benchmark, while the simple baselines are sub-millisecond to millisecond scale. The cost-linear baseline is cheaper but optimizes a different tradeoff and has lower total utility.

Figure 2: **Narrow positive result, not a general victory.** Compitum's synthetic advantage is utility-per-dollar at zero constraint violations under the stated benchmark conditions, with an explicit latency cost.

5 Empirical Reading (the honest part)

E1 — Cost-efficiency at zero violations, with a latency cost (synthetic). On a synthetic routing benchmark (`bench_summary.csv`; cases A–D; strategies `compitum` vs `fixed_best`, `greedy`, ε-decay, UCB1, Thompson, `cost_linear`), Compitum attains higher utility-per-dollar than the fixed and greedy baselines (e.g. case A: ≈ 0.96 vs ≈ 0.82) at *zero* constraint violations. The honest cost: its $p95$ latency is $\sim$6 s versus sub-millisecond for the trivial baselines—the geometric scoring and constraint solve are not free. The claim is therefore narrow: *when constraint-correctness and cost-efficiency matter more than raw decision latency*, the geometry pays for itself. This is reproducible from the repository and is falsified if a rerun shows no utility-per-dollar advantage or any constraint violation.

E2 — The geometric advantage does not transfer to flat data (a prediction, not a failure). On engineered-feature materials data (UCI superconductivity; a matbench demo) the curved-metric advantage *disappears*: fitting suppression-vs-distance scaling near top performers yields exponent $\alpha \approx 0.9$–1.2 (linear, not quadratic), with the learned metric performing comparably to Euclidean, and gradient norms that do not approach zero under refinement. We read this not as a defeat but as a *boundary condition the framework predicts*: a metric is only meaningful on the right chart. Engineered descriptors are a warped embedding, not the system's control knobs; absent a stationary point in that chart, the linear term dominates and geometry $\approx$ Euclid. The method is expected to help where a trusted low-dimensional control manifold exists, and *not* to manufacture curvature where the data are flat. Stating where a method fails, and why, is the part most papers omit.

6 Limitations and Program Direction

We are explicit about the gap between what is proved and what is hoped.

- **Local, not global.** Section 3 gives non-increase under bounded, fixed-selection assumptions; global asymptotic stability (radially unbounded potential, single equilibrium, switching dynamics) is open.
- **Surrogate, not first principles.** F_s is a designed scalarization; we do not claim it is the unique or correct potential. Misspecification weakens behavior.

- **Stability $\neq$ safety.** A stable router can be confidently wrong; external safety constraints remain necessary.
- **Curved-space validation is future work.** The bounded-observer/curvature thesis (shared with *The Wicked Prior* and the SRMF notes) is validated here only at the level of runnable invariants, not as a closed physical theory. Where it goes: a continuous-time control Lyapunov grounding, and validation on systems with genuine control manifolds.

7 Relation to the Program

This artifact is one bloom of a larger program organized by a single commitment—*adequacy should be measurable, not asserted. The Hypothesis Surface* states the epistemology; *Come* operationalizes it at the level of test suites (mutation-tested seams between reasoning systems); *The Wicked Prior* and the SRMF/geometric-systems notes carry the bounded-observer geometry into narrative and physics. Compitum is the routing instance: a place where the commitment is cheap to check, because the checks are code.

8 Reproduction

Clone the repository and run the falsification harness (`pytest -q tests/invariants`); regenerate the synthetic benchmark and inspect `bench_summary.csv`; the materials negative (E2) is reproduced by the `examples/supercon` scripts. Every numeric claim above is intended to be re-derivable; where it is not yet turnkey, that is a defect of this draft, not a feature.

Conclusion

We have argued for routing whose behavior you can check rather than take on faith, given the assumptions and lemmas under which Compitum's energy is Lyapunov-style stable, mapped each property to a test designed to break it, and reported an honest empirical picture including a negative result. In the spirit of *Come*, this is not a finished proof of a finished system. It is a substrate, offered incomplete on purpose, for whoever wants to run it, doubt it, and extend it.

Acknowledgments and Further Reading

The references are deliberately limited to freely accessible sources, so any reader can verify every claim here at no cost—the point of submitting to the noosphere. The author's understanding is nonetheless indebted to the standard literature studied in preparation for AGI-26, ICML, and NeurIPS—among them Khalil's *Nonlinear Systems*, Bhatia's *Positive Definite Matrices*, and Amari's *Information Geometry and Its Applications*—which readers with library access will find deepen every section. Citing the open sources is a choice about access, not a comment on the value of the canon.

All references below are freely accessible online — no paywall, nothing to buy.

References

[1] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008. Full text freely available: https://fbsbook.org. (Lyapunov and LaSalle stability.)

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. Full text freely available: https://web.stanford.edu/~boyd/cvxbook/. (Positive-definite matrices; Lagrange duality and shadow prices.)

[3] P. Tiffany. The Wicked Prior as a Bounded-Observer Manifold: Atlas Fracture, Stackelberg Parentage, and Grace-Flow Repair. 2025. Open access: https://paultiffany.github.io/wicked-geometry/.