Origin Axiom — Phase 3 (Exploratory Add-on): Flavor Phase Integration

Origin Axiom Collaboration

Abstract

Phase 3 of the Origin-Axiom program is a ledger-compatible flavor-sector calibration add-on. Under a fixed, explicitly stated ansatz, and for a frozen snapshot of CKM and PMNS CP-phase targets, we scan a one-parameter family $\theta \in [0, 2\pi)$ to extract a best-fit $\hat{\theta}$ and an uncertainty interval. We expose this fit as a schema-stable θ-filter JSON artifact, designed to be consumed by the Phase 0 corridor ledger alongside the existing Phase 1 and Phase 2 filters. We then inject the fitted θ into the Phase 2 vacuum-residue mechanism to compute a diagnostic $\Delta\rho_{\text{vac}}(\theta)$ curve, keeping the injection strictly one-way (no feedback of Phase 2 into the flavor fit). In the baseline configuration documented here, the Phase 0 ledger reports that the combined corridor is empty once the Phase 3 filter is applied. We interpret this as a negative test for the locked Phase 3 ansatz/target combination, not as a proof that no Origin-Axiom-compatible corridor exists. The purpose of this paper is to make this calibration step and its failure mode reproducible, auditable, and upgradeable as flavor data and ansatz choices evolve.

1 Introduction

1.1 Role of Phase 3 in the program

Phase 3 is a *ledger-compatible add-on*: it extracts a candidate value and uncertainty interval for θ by fitting a fixed, explicitly stated ansatz to frozen external flavor-sector targets (CKM and PMNS CP-phase information). The Phase 3 deliverable is not the internal fit machinery; it is the exported, schema-stable artifact `phase_03_theta_filter.json`, designed to be consumable by the Phase 0 corridor/ledger aggregation without requiring Phase 3 internals.

Concretely:

- **Inputs**: external target snapshot encoded in `phase3/fit/targets.yaml` (treated as frozen for this Phase 3 run) and the fixed ansatz implementation.
- **Outputs**: (i) fit summary and diagnostics tables, (ii) figures, (iii) this PDF report, and (iv) the ledger artifact `phase_03_theta_filter.json`.

1.2 What Phase 3 does and does not establish

Phase 3 establishes only the following narrow fact: under the Phase 3 baseline ansatz and frozen targets, there exists a best-fit θ and an uncertainty interval produced by an explicit, reproducible scan.

Non-claims (explicit). Phase 3 does *not* claim:

- a derivation of a unique fundamental constant $\theta_{\star}$;
- a proof that the Origin Axiom is correct, complete, or uniquely determined;
- a reduction of Standard Model free parameters, or a predictive theory of the CKM/PMNS structures;
- a measurement-level global fit (this is not a replacement for PDG/NuFIT analyses);
- robustness to arbitrary re-parameterizations or model classes beyond the declared ansatz family.

1.3 Success criteria (Phase 1/2 style contract)

A Phase 3 run is considered successful iff:

- the workflow gate reproduces all declared artifacts end-to-end;
- the bundle manifest lists exactly the declared inputs/outputs with hashes and sizes;
- the paper claims remain within the narrow scope above (no implied derivation or discovery).

Ledger-level outcome. At the level of the Phase 0 corridor ledger, Phase 3 contributes only a filter: it may further narrow the corridor defined by earlier phases or, in some configurations, eliminate it entirely. In the baseline configuration documented here (ansatz locked in `phase3/fit/targets.yaml` with the stated CKM/PMNS snapshot), the Phase 0 ledger reports that the combined corridor is empty once the Phase 3 filter is applied. We interpret this as a negative result for that specific ansatz/target combination, not as a proof that no Origin-Axiom-compatible corridor exists.

2 Flavor-phase fit pipeline

This section describes the Phase 3 "methods" in the narrow, empirical sense: how we turn frozen external flavor targets into a one-parameter scan over θ and a machine-readable θ-filter.

2.1 External targets and ansatz

The fit is anchored to a snapshot of CKM and PMNS CP-phase information, encoded in `phase3/fit/targets.yaml`. For the present run we treat this file as fixed input: it records the numerical values taken from standard references (PDG and NuFIT) and tags them with a simple "ansatz label" identifying the mapping family used in Phase 3.

We do not attempt to explore a large model space. Instead we fix a single, explicit ansatz that maps the Origin-Axiom phase parameter θ to the relevant flavor-sector phases. The choice is pragmatic and clearly versioned in `phase3/fit/ANSATZ_CONTRACT.md`; it is not claimed to be unique or fundamental. At the level of this paper it is sufficient to regard the ansatz as a smooth map

$$\theta \longmapsto \big(\delta_{\mathrm{CKM}}(\theta),\, \delta_{\mathrm{PMNS}}(\theta), \dots\big),$$

with the explicit parameterization and any discrete offset hypotheses versioned in `phase3/fit/ANSATZ_CONTRACT.md`. This keeps the ledger-facing artifacts agnostic to low-level implementation details while making the mapping auditable.

2.2 Objective function and scan

Given a candidate value of θ, the ansatz produces predicted flavor phases. We compare these to the external targets using a simple $\chi^2(\theta)$ objective. The present implementation evaluates χ^2 on a regular grid in $\theta \in [0, 2\pi)$ and records, for each grid point,

- the value of θ,
- the corresponding $\chi^2(\theta)$, and
- any auxiliary diagnostics used in the paper figures.

From this grid we extract a best-fit value $\hat{\theta}$ (the minimum of χ^2) together with a one-dimensional uncertainty interval defined by a simple $\Delta\chi^2$ criterion. The grid resolution, the precise convention for $\Delta\chi^2$, and the resulting numerical values of $\hat{\theta}$ and its interval are recorded in the accompanying fit artifacts:

- the summary table `phase3/outputs/tables/theta_fit_summary.csv`;
- the diagnostics table `phase3/outputs/tables/theta_fit_diagnostics.json`.

We treat these files as part of the formal methods contract: any future re-implementation of the Phase 3 fit must reproduce their contents (within numerical tolerance) for the same ansatz/targets configuration.

2.3 Fit artifacts

The primary artifacts of this section are:

- the summary table `phase3/outputs/tables/theta_fit_summary.csv`;
- the diagnostics table `phase3/outputs/tables/theta_fit_diagnostics.json`;
- the figure `phase3/outputs/figures/fig1_theta_fit_diagnostics.pdf`; and
- the ledger-facing θ-filter `phase3/outputs/theta_filter/phase_03_theta_filter.json`.

These are the objects that Phase 0 and the other phases are allowed to depend on; the internal fit code is treated as an implementation detail, provided it reproduces the same artifacts under the gate described in the reproducibility appendix.

3 Injection into the Phase 2 vacuum-residue mechanism

This section explains how the Phase 3 best-fit phase $\hat{\theta}$ is used as an input to the Phase 2 vacuum-residue machinery, and what the resulting diagnostics mean.

3.1 One-way injection hook

Phase 2 implements a vacuum-residue mechanism and exposes a one-parameter injection hook for a phase-like parameter. In Phase 3 we treat this as an opportunity to test whether the empirically fitted flavor phase is at least compatible with the vacuum-side construction.

Operationally, we:

1. take the same θ grid used in the fit;
2. pass each θ value through the Phase 2 injection hook, keeping all other Phase 2 settings fixed at their baseline values; and
3. record the resulting residue metric $\Delta\rho_{\mathrm{vac}}(\theta)$.

The injection is strictly one-way: the Phase 2 outputs are not fed back into the flavor fit, and no attempt is made to re-optimise θ on the basis of vacuum-side information.

3.2 Diagnostic curve and interpretation

The resulting curve $\Delta\rho_{\mathrm{vac}}(\theta)$ is plotted in `phase3/outputs/figures/fig2_delta_rho_vac_vs_theta.pdf`, with metadata in `phase3/outputs/figures/fig2_delta_rho_vac_vs_theta.meta.json`. In this paper we use the curve purely as a qualitative diagnostic: it shows how the Phase 2 residue responds as θ is moved through the corridor defined by the flavor fit, but we do not claim any quantitative prediction for cosmological observables.

The only downstream interface that is allowed to act on Phase 3 is the ledger-facing θ-filter described in Appendix D, which compares the Phase 3 admissible region with the existing Phase 0–2 corridor. The injection described in this section is therefore a "sanity check" and a source of intuition, not a binding constraint.

4 Falsifiability and Failure Modes

Phase 3 is falsifiable in the narrow sense appropriate to an exploratory fit: the mapping from θ to the frozen flavor targets can fail to produce a stable, interpretable interval.

4.1 Explicit failure conditions

We treat the Phase 3 extraction as *invalid* if any of the following occur:

- **Non-identifiability:** multiple disjoint θ regions yield indistinguishably good minima (flat or multi-modal χ^2 without a defensible interval).
- **Offset ambiguity:** the discrete offset sweep yields competing hypotheses with comparable minima, preventing a pre-registered baseline choice from being meaningful.
- **Target drift collapse:** a reasonable update of the external targets (future PDG/NuFIT releases) removes the minimum or shifts it outside any stable corridor overlap with other phases.
- **Ansatz fragility:** small, clearly stated modifications to the mapping family (within the declared ansatz class) destroy the existence of a stable minimum, indicating the result is an artifact of parameterization rather than a durable extraction.
- **Ledger incompatibility:** when the Phase 3 θ-filter is injected into the Phase 0 corridor ledger, the resulting combined corridor becomes empty (no θ survives simultaneous application of the Phase 0–2 and Phase 3 filters). This is treated as evidence against the locked Phase 3 ansatz/target combination until the ansatz or external targets are revised.

4.2 What Phase 3 can support downstream

If Phase 3 passes the checks above, it supports only this downstream use: the exported `phase_03_theta_filter.json` interval can be compared by Phase 0 ledger tooling to other phase constraints to test whether a corridor overlap emerges. No stronger inference is warranted.

5 Discussion and Limitations

5.1 Summary of the Phase 3 outcome

Phase 3 is designed as an exploratory, ledger-compatible calibration step: it uses external flavor-sector information (CKM and PMNS CP-phase data) to extract a one-parameter fit for θ, exports the resulting interval as a ledger-facing θ-filter artifact, and then probes the Phase 2 vacuum-residue mechanism at and around that interval.

In the baseline configuration documented here, the Phase 3 fit pipeline (Section **??**) produces a best-fit value $\hat{\theta}$ and an associated uncertainty interval derived from a one-parameter scan under a fixed ansatz and frozen target snapshot. This interval is encoded in the artifact `phase3/outputs/theta_filter/phase_03_theta_filter.json`, which conforms to the Phase 0 corridor interface and underpins claim C3.1.

When this artifact is injected into the Phase 0 corridor ledger alongside the Phase 0–2 filters, the resulting combined corridor is empty: no θ survives the simultaneous application of all filters. This is the central negative result of the present Phase 3 run. By construction, we interpret this as evidence against the locked Phase 3 ansatz/target combination (and, to a lesser extent, against the present alignment between Phases 2 and 3), not as a global falsification of the Origin-Axiom hypothesis.

5.2 Dependence on external flavor data

The Phase 3 extraction is conditional on a frozen snapshot of external flavor data:

- The targets and their uncertainties are encoded in `phase3/fit/targets.yaml`, with provenance ties to contemporary global fits (PDG, NuFIT, etc.).
- The present paper treats this snapshot as immutable for the purposes of the reported run; any update to the external data requires a new Phase 3 run with a new bundle and potentially a different outcome.
- Correlation structures, hierarchy assumptions, and other simplifications are explicitly documented in the configuration and should be revisited as global flavor analyses evolve.

In particular, the existence and location of $\hat{\theta}$ and its interval are sensitive to target updates. One of the explicit failure conditions in Section 4 is a "target drift collapse," in which future data remove or significantly displace the minimum relative to the Phase 1/2 corridor. Users of the Phase 3 filter must therefore treat the present corridor intersection (or its absence) as provisional, tied to this one external snapshot.

5.3 Ansatz fragility and model incompleteness

The ansatz used in Phase 3 is intentionally modest: it is a one-parameter family that maps θ into a handful of CP-sensitive flavor observables, with a small discrete set of offset hypotheses. This serves the goal of defining a clean, reproducible test bed, but it also imposes strong limitations:

- The ansatz is not a realistic model of the CKM/PMNS matrices and does not attempt to reproduce the full structure of modern flavor fits.
- The discrete offset family is very restricted; we presently select $b_{\mathrm{PMNS}} = \pi$ as the baseline because it minimizes the fit objective among the candidates listed in Appendix **??**, but a richer ansatz space could easily produce different minima.
- Small, controlled modifications within the declared ansatz class may destroy the stability of the minimum or significantly change the extracted interval. This fragility is explicitly listed as a falsifiability condition in Section 4.

For these reasons, the negative corridor result should not be over-interpreted: it says that one particular slice through flavor-parameter space, under a very specific ansatz, is difficult to reconcile with the present Phase 1/2 corridor. It does not rule out other ansatz choices, other ways of embedding flavor information, or revisions to the Phase 2 mechanism itself.

5.4 Corridor-level interpretation

The Phase 0 corridor ledger provides a technical framework for combining θ-filters across phases. Within this framework, the Phase 3 result plays three roles:

- It supplies a filter that is structurally comparable to the Phase 1 and Phase 2 filters, enabling a clean intersection test.
- It provides an explicit example of a configuration in which the combined corridor becomes empty, illustrating that the ledger can record negative results as first-class outcomes.

- It anchors the procedural falsifiability claim (C3.3): an empty corridor under the locked ansatz/target combination is treated as evidence against that configuration, and any future Phase 3-like attempt must state in advance how such outcomes will be interpreted.

From a programmatic point of view, the empty corridor is useful even if it is disappointing: it identifies a concrete mismatch between one tractable flavor-encoding and the current residue-based corridor. That mismatch can then be targeted by future work.

5.5 Outlook

The present Phase 3 run is not intended to be the final word on flavor-sector integration. Instead, it establishes a reproducible template for how such attempts should be structured and recorded:

- **Upgrading the ansatz.** Future iterations can enlarge the ansatz family, add more realistic parameterizations, or explore different ways of coupling θ into the CKM/PMNS structures. Each such change should be documented in an updated `ANSATZ_CONTRACT.md` and produce a new, clearly labelled Phase 3 bundle.
- **Updating external data.** As global flavor fits evolve, new snapshots can be encoded in `targets.yaml` and used to rerun the Phase 3 pipeline. Comparison across snapshots may reveal trends in how the Phase 3 corridor constraint moves relative to Phase 1/2.
- **Refining downstream diagnostics.** The present injection into Phase 2 uses a single diagnostic curve $\Delta\rho_{\mathrm{vac}}(\theta)$. Future work may add more refined diagnostics, or connect the injected θ region to Phase 2 observables in different ways.
- **Exploring Phase 4 and beyond.** If the Origin-Axiom program develops additional phases (for example, more direct observational tests), the Phase 3 template provides a blueprint for how to expose their filters to the corridor ledger and how to declare their claims and non-claims.

In short, the Phase 3 paper should be read as an explicit, negative but constructive experiment: it shows how one plausible attempt to connect a flavor-calibrated θ to the existing corridor fails, and it records enough detail that the attempt can be repeated, modified, or superseded in a controlled way.

A Claims Table (Phase 3)

See `phase3/CLAIMS.md` for the live evidence map.

B Reproducibility

Phase 3 follows the same gate structure as Phases 0–2, with explicit levels and a single Snakemake entry point. The authoritative, machine-readable reproducibility contract is documented in `phase3/REPRODUCIBILITY.md`, but we summarize the structure here.

Levels.

- **Level A (repo snapshot):** verify the paper and bundle from an existing repository snapshot. This level checks that the committed artifacts and the bundle manifest are internally consistent. Entry point: `bash scripts/phase3_gate.sh --level A`.
- **Level B (regenerate artifacts):** rebuild all tables, figures, the paper, and the paper bundle from source using Snakemake, then verify the bundle. This is the default reproducibility target for external readers: `snakemake -s phase3/workflow/Snakefile -c 1 all`, followed by `bash scripts/phase3_gate.sh --level B`.
- **Level C (heavy runs):** optional developer mode that can populate `phase3/outputs/runs/` (git-ignored) with additional scans or diagnostics. This mode is not required to reproduce any claims in this paper, and is intended only for extended exploration: `bash scripts/phase3_gate.sh --level C`.

Bundle and evidence location. All artifacts used as evidence for the Phase 3 claims are collected under `phase3/outputs/paper_bundle/`, with a `run_index.json` and `bundle_manifest.json` that enumerate inputs and outputs with hashes and sizes. The canonical PDF for this phase is `phase3/artifacts/origin-axiom-p` built by the Phase 3 Snakemake workflow.

Table 1: Discrete fixed-offset sweep for b_{PMNS} with a single fitted parameter θ. Lower χ^2 is better; $\Delta\chi^2$ is relative to the best row.

b_{PMNS} (deg)	θ^* (deg)	χ^2	$\Delta\chi^2$	$\theta^{\mathrm{lo}}_{68\%}$ (rad)	$\theta^{\mathrm{hi}}_{68\%}$ (rad)
180	65.682	0.675474	0.000000	1.120292	1.172128
90	65.790	1.881895	1.206421	1.122491	1.174013
-90	65.556	9.093410	8.417936	1.118407	1.169929
0	65.916	12.712767	12.037293	1.124690	1.176212

References

C Phase 3 θ-filter artifact (Phase 0 ledger interface)

Phase 3 emits a machine-readable θ-filter artifact: `phase3/outputs/theta_filter/phase_03_theta_filter.json`. This file is the Phase 3 contribution to the Phase 0 corridor method: it reports an admissible set of θ values under an explicitly declared Phase 3 test suite, along with provenance sufficient for audit and reproduction.

C.1 Declared test suite

Phase 3 is an empirical calibration filter (not an OA-binding demonstration). We therefore define a single test:

- **fit_compat_interval:** θ is admissible if it lies within the declared fit interval reported in `theta_fit_summary.csv`.

This choice is intentionally conservative: it makes the corridor definition explicit and reproducible, and it avoids parameter proliferation or implicit re-optimization beyond the one-parameter scan.

C.2 Schema compliance

The JSON conforms to the Phase 0 ledger interface: it declares a θ domain on $[0, 2\pi)$, provides an interval-form corridor representation, and includes a deterministic grid+pass array representation, plus provenance bindings (commit/config/environment hashes when available).

C.3 Ledger outcome in the current configuration

The Phase 0 ledger applies the Phase 3 θ-filter on top of the existing Phase 0–2 filters and records the resulting combined corridor in `phase0/phase_outputs/theta_corridor_history.jsonl`. In the baseline configuration documented here, the ledger reports that the combined corridor is empty once the Phase 3 filter is applied. This is logged as a negative test for the present Phase 3 ansatz/target combination, and it is intentionally encoded so that future runs with alternative ansatz choices or updated external targets can be compared against this outcome.