Gap-Free Rigidity, Robust Certificates, and Chamberwise Global Cover/Atlas Universality for Extremal Growth in Finite Positive Templates

Drew Alexander Higgins

February 21, 2026

Abstract

We present a gap-free rigidity mechanism for extremal (minimal) growth in finite families of positive template operators. The core outcome is a finite certificate dichotomy: for every reduced descriptor instance, either one produces a concrete obstruction witness (OG1–OG4), or one produces a canonical Doeblin witness together with a finite recurrence gate (MB–RC1 in canonical multi-block mode; RC1 in the single-block special case) that yields uniform projective contraction along all tight trajectories. We then upgrade the pipeline to a robust form: each certificate is accompanied by explicit perturbation margins and a computed stability radius ε^* so the decision is stable under small changes of template entries. The resulting package yields a deterministic, checkable route from reduction to quantitative Hilbert-metric contraction, with constants that can be read off from the certificate itself. Finally, on compact normalized parameter families (within the certified class) we obtain a witness-producing global cover/atlas decider: either a finite set of robust certificates covers the region with a single uniform robustness radius and uniform rate pack (good-global contraction universality), or the procedure returns explicit obstruction/degeneracy data locating where and why uniform contraction cannot hold, packaged as a boundary-atlas report.

1 Overview

This paper is built for two complementary goals.

Minimal mental model. A finite family of positive templates induces a reduced tight core capturing all extremal (minimal-growth) behavior. Either there is a finite obstruction witness (OG1–OG4), or a canonical Doeblin witness on an aperiodic tight component together with a finite recurrence gate (MB–RC1 in canonical multi-block mode; RC1 in the singleton special case) yields uniform projective contraction along all tight trajectories. Robust certificates attach explicit margins and a computed stability radius $\boldsymbol{\varepsilon}^*$, so the conclusion is stable under small perturbations. On any compact normalized parameter family Θ within the certified class, this openness becomes a finite cover: the outcome is always *cover-or-atlas*—either a single uniform good-global contraction certificate, or a boundary-atlas report localizing the degeneracy where uniform contraction cannot hold.

- **Structural goal:** explain why "optimality forces regularity" in finite positive-template growth problems.
- **Operational goal:** give a finite certificate contract that either returns a specific obstruction (OG1–OG4) or returns quantitative contraction data that can be verified directly.

1.1 Certified class and scope contract

All "decidability" and "global universality" claims in this paper are made *inside a certified class* of finite positive-template models. The contract has two parts.

- **Finite positive-template data.** The input is a finite reduced graph with nonnegative template blocks and reduced-cost data (Definition 1.2). All subsequent objects (tight core, SCCs, Boolean-pattern Doeblin automaton, avoidance automaton, and witness payloads) are finite.
- **Structural zeros are part of the model.** On parameter families, perturbations are treated as *support-preserving* whenever the model declares a fixed sparsity pattern for each template (Definition 3.1). This is the natural regime for most finite-template constructions (legal/illegal couplings), and it is exactly what makes "NO" outcomes robust via support-stability (Section 25.2).

A canonical compact normalized family Θ_{can} satisfying this contract is recorded in Section 3. Outside this certified class, we do not claim decidability of the full joint spectral radius problem; instead, we locate an explicit frontier where witness-producing decisions and robustness are available.

Technically, the contraction step is a quantitative use of Hilbert's projective metric and Birkhoff's contraction coefficient [10, 11, 12, 13]. Conceptually, extremal itineraries are modeled by a finite-type shift and its sofic refinements [17], and the resulting rate packs interface naturally with the language of linear cocycles, random matrix products, and Lyapunov exponents [14, 15, 16]. The motivating growth viewpoint aligns with transfer-matrix product growth systems in statistical mechanics and related positive-operator settings [18].

The technical spine proceeds in three layers.

- **Reduction and tight core:** the descriptor reduction produces a tight subgraph $G^{(0)}$ and local face data.
- **Gap-free forcing and contraction:** calibrated minimizers concentrate on the tight core; slack and return-cost arguments yield positive-density forcing; Doeblin and recurrence then give projective contraction on minimizers.
- **Certificates and robustness:** each step is packaged as a finite certificate with explicit robustness margins.

1.2 Positioning, novelty, and what is actually proved

The organizing theme is that *optimality induces projective regularity* once the dynamics are reduced to a tight core. In broad switching/JSR settings, one cannot generally decide stability, and one should not expect a single universal mechanism. The point here is different: within a certified finite positive-template class, the paper gives a *complete, finite, witness-producing* alternative.

How the standard ingredients are used. The projective contraction step is classical: strict positivity on a face gives a Birkhoff contraction factor in Hilbert metric [10, 11, 12]. The novelty is not that implication. The novelty is that the paper makes the *presence* of that implication decidable by finite witnesses and then makes it *robust* under perturbation. Symbolic-dynamics language (SFT/sofic shifts) is used as the natural bookkeeping for the tight language and its refinements [17]; this aligns with the cocycle viewpoint common in random matrix products and

Lyapunov exponent theory [14, 15, 16]. The transfer-matrix perspective provides an intuition for why nonnegative template products are a natural substrate [18], and the Perron–Frobenius growth viewpoint for nonnegative operators is classical [13].

What is new (inside the certified class). The contributions are packaged so that each claim is backed by a finite artifact.

Contribution	What is produced and why it is different
Witness-producing decidability frontier (OP–JSR–RU)	A deterministic procedure that returns either a robust contraction certificate (Doeblin word + recurrence modulus + explicit rate pack + robustness radius) or a finite obstruction witness (OG1–OG4), including an explicit avoidance cycle when recurrence fails. This turns "is there a robust uniform contraction mechanism?" into a decidable question with verifiable witnesses.
Finite cover-or-atlas completion on parameter families (OP–ATLAS–FIN)	On compact normalized families within the certified class, region mode exports a single `FrontierAtlasReport` that stratifies the region by the minimal failing gate and attaches a carry-forward decision radius per chamber. The artifact is batch-verifiable (VerifyGlobalReport).
Effective extremal itinerary model (OP–MATHER–EFF)	The extremal language is made explicit: a computable SFT carrier for tight dynamics plus a sofic refinement driven by the Doeblin/avoidance mechanism. This yields an exportable `MatherSoficModel` object whose content is decided by finite graph predicates.

What is not claimed. All decidability and robustness statements are made *only* inside the certified positive-template class. In particular:

- the paper does not claim decidability of joint spectral radius in unrestricted families (which is consistent with known undecidability barriers [1]);
- the paper does not claim that every positive family admits a Doeblin+recurrence contraction mechanism; rather, it returns explicit witnesses showing *why* such a mechanism fails (OG1–OG4);
- region-level statements remain conditional on the certified class contract, including support-preserving perturbations when structural zeros are declared.

Why this is the "frontier" result. The theorems do not merely prove a stability statement when assumptions hold. They give a *complete decision object* (certificate or obstruction) together with a certified local invariance radius on both the YES and NO sides whenever the fixed-support contract applies. This is the sense in which the work isolates a decidable island and exports it as reusable artifacts.

1.3 Main results at a glance and how to verify them

The paper exports three primary theorem–artifact pairs. Each pair is designed so a reader can check the claim by inspecting a finite witness object and running finite graph predicates.

Theorem 1.1 (Master robust universality dichotomy)**.** *Within the certified operator class (Definition 1.2), the decision pipeline returns* exactly one *of the following finite outcomes.*

- *A `RobustCertificate` whose fields certify: (i) a tight aperiodic core and consistent active face, (ii) a canonical Doeblin witness and (optionally) a canonical Doeblin family $\mathcal{B}(H)$ (Section 12 and Section 13), (iii) a finite recurrence gate in combinatorial mode—either single-block **(RC1)** for the witnessed word, or multi-block shell rigidity **(MB–RC1)** for the family—together with an explicit bounded-gap modulus, and hence (iv) an explicit projective contraction rate package with a robustness radius.*
- *An `ObstructionWitness` labeled OG1–OG3, certifying failure of (i) or (ii) by a finite predicate.*
- *An `ObstructionWitness` labeled OG4, certifying failure of the chosen recurrence gate by a finite directed-cycle witness in the appropriate avoidance graph (single-block $OG4(H, w)$ or multi-block $OG4^{\mathrm{MB}}(H)$); in particular, the obstruction can be realized by a tight minimizing itinerary avoiding the certified contraction block(s).*

Consequently, RU-Stable *holds if and only if the pipeline does* not *output an obstruction witness.*

Theorem-level claim	**Meaning (inside the certified class)**	**Exported object**
Decidable frontier (OP–JSR–RU)	DecideRUStable returns exactly one finite witness: either a robust contraction certificate, or an obstruction witness OG1–OG4; moreover RU-Stable is equivalent to $\neg(\mathrm{OG1} \vee \cdots \vee \mathrm{OG4})$. (Theorems 1.4 and 1.5)	`RobustCertificate` or `ObstructionWitness`
Atlas universality (OP–ATLAS–FIN / OP–FRONTIER–MAP)	On compact normalized families within the contract, region mode terminates with a single exported `FrontierAtlasReport` that stratifies by the minimal failing gate and attaches a carry-forward decision radius per chamber. (Theorems 22.3 and 35.9)	`FrontierAtlasReport` (embedded in `GlobalUniversalityReport`)
Effective extremal language (OP–MATHER–EFF)	The extremal itinerary structure admits a finite symbolic model: a computable SFT carrier plus a sofic refinement driven by the Doeblin/avoidance mechanism, with a sharp *multi-block* shell-rigidity dichotomy (MB–RC1 vs OG4). (Theorem 15.3)	`MatherSoficModel`

Verification guide (artifact-first). The exported objects are meant to be checkable without reconstructing the full narrative. At a high level:

- **For a contraction certificate:** verify tight SCC/face validity (Section 11), verify the Doeblin word by Boolean-pattern multiplication (Algorithm 6), verify recurrence by cycle detection in the appropriate avoidance graph (Algorithm 7 for single-block RC1, or the multi-block avoidance check of Section 13 for MB–RC1), then read off the explicit rate pack (Section 18) and robustness radius (Sections 25–28).

- **For an obstruction witness:** verify the finite predicate corresponding to OG1–OG4 directly from the witness payload (Section 16 and the `ObstructionWitness` schema in Section 23).
- **For a region atlas:** validate the JSON schema, then run VerifyGlobalReport (Algorithm 10) which calls VerifyFrontierReport chamberwise and checks the chamber-level carried radii.

Why this is referee-friendly. The decisive steps are finite. Even when the underlying dynamics are framed analytically (projective contraction), the presence/absence of the needed mechanism is decided by explicit finite automata built from the tight language and declared face supports.

Where to find what

If you are looking for	See
Decidable-island framing and scope limits	§1.4 and §1.6
Reduction, normalization, and the induced tight-core boundary system	§2 and §4 (with computation in §5)
Obstruction taxonomy (OG1–OG4) and witness objects	§16 and §33
Doeblin-word decision and witnessed word extraction	§12
Recurrence gates, forcing, and syndetic tools (single-block and multi-block)	§14 and §13
Effective extremal itinerary model (sofic Mather package)	§15 and §24
Contraction from Doeblin plus recurrence (including disjoint-occurrence counting in RH2)	§17
Rate-pack extraction (constants, decay rates, and bounds)	§18
Robustness radii and the stability layers (Robust Certificate Contract to Recurrence Stability)	§25, §26, §27, §28
Global-on-Θ cover/atlas guarantee and output objects	§30, §31, §32, and §32.4 (with the Θ-family contract in §3)

1.4 OP–JSR–RU: witness-producing decidability frontier for robust switching stability

In the general theory of switching systems and joint spectral radius (JSR), many natural decision problems are not decidable in full generality. For instance, even boundedness questions for products of a fixed finite matrix set can be undecidable in broad rational classes [1]. Accordingly, progress on "where stability is decidable" typically takes the form of identifying structurally constrained subclasses in which stability reduces to finite witnesses.

The certificate program developed here provides such a subclass. Rather than attempting to decide *JSR* in complete generality, we decide a stronger, certificate-based property: whether the instance admits a *robust uniform contraction mechanism* that is both checkable and stable under perturbation.

Definition 1.2 (Certified positive-template instance). *A* certified instance *is an input datum for the pipeline contract (Section 20) consisting of*

- *a finite reduced directed graph $G = (V, E)$,*
- *a nonnegative template (or fixed-length block) $T_e \in \mathbb{R}_+^{d\times d}$ attached to each edge $e \in E$,*
- *a calibrated potential ϕ and reduced costs $c_\phi(e) \geq 0$ defining the tight edge set $E^{(0)} = \{c_\phi = 0\}$,*
- *canonical support sets $S(v) \subseteq \{1, \ldots, d\}$ for descriptor states.*

All derived objects used by the decision procedure (tight SCCs, face maps, Doeblin search automata, avoidance graphs, and robustness margins) are computed from this finite data.

Definition 1.3 (Robust contraction certificate property (RU-Stable)). *Fix a perturbation model and norm as in Section 25. A certified instance satisfies* RU-Stable *if the pipeline returns a* robust contraction certificate *whose fields include*

- *a tight aperiodic SCC H on which the tight-core data are valid,*
- *a Doeblin word witness w with margin $\delta > 0$ on the SCC face support $S(H)$,*
- *a recurrence modulus (RC1) for (H, w), yielding bounded-gap recurrence of w along every tight minimizer supported in H,*
- *an explicit projective contraction-rate pack (τ, C, κ),*
- *an explicit robustness radius $\varepsilon^* > 0$ such that all of the above remain valid (with explicit degraded constants) for perturbations of size at most ε^*.*

Decision procedure. Define DecideRUStable to be the pipeline of Section 20 restricted to the fully finite branch logic through OG4. On input a certified instance, it computes the tight-core objects and returns exactly one artifact:

- a robust contraction certificate (hence RU-Stable), or
- a finite obstruction witness in OG1–OG4 (hence not RU-Stable by the certificate rules).

Theorem 1.4 (Witness-producing decidability frontier). *For every certified instance (Definition 1.2),* DecideRUStable *terminates and returns exactly one of the following finite, verifiable artifacts.*

- *A* finite obstruction witness *in OG1–OG4 (Section 20), certifying that the data required by Definition 1.3 cannot be obtained from the instance by the certificate rules. In particular, OG4 returns an explicit directed cycle in the avoidance automaton, producing a tight periodic trajectory that avoids the Doeblin word forever.*
- *A* robust contraction certificate *establishing* RU-Stable *for the instance, including an explicit robustness radius ε^* and explicit rate degradation formulas.*

Consequently, membership in RU-Stable *is decidable on the certified positive-template class, and the decision is witness-producing.*

Proof. Termination. All constructed objects are finite. Tight-edge extraction, SCC decomposition, and face-map checks are finite graph computations. Doeblin-word existence is decided by reachability in a finite Boolean-pattern product automaton (Algorithm 6). RC1 is decided by cycle detection in the finite avoidance graph $\mathcal{A}_{\neg w}(H)$ (Algorithm 7). Therefore the procedure halts.

Exhaustiveness and exclusivity. The branch logic in Section 20 partitions outcomes into mutually exclusive cases OG1–OG4 and the contraction-certificate branch. Each case is determined by a finite predicate and its negation (aperiodic SCC existence, face-map validity, Doeblin reachability, avoidance-cycle existence). Hence exactly one branch is taken.

Soundness of obstruction witnesses. OG1 certifies the absence of aperiodic tight SCCs, so the first required field of Definition 1.3 cannot be met. OG2 returns a finite inconsistency cycle demonstrating that the SCC face support is not invariant under the tight dynamics, so a face-based Doeblin and contraction statement is not well-posed under the certificate contract. OG3 certifies that no Doeblin word exists on $S(H)$ (via a cut witness or a product-automaton non-reachability certificate), so no certificate can supply a Doeblin margin $\delta > 0$. OG4 returns a directed cycle in $\mathcal{A}_{\neg w}(H)$, yielding a tight periodic trajectory that avoids the Doeblin word forever (Proposition 16.4); therefore bounded-gap recurrence of w (RC1) fails, and no uniform contraction statement of the certified form can hold.

Soundness of the certificate branch. If the procedure returns a contraction certificate, then by construction it includes a Doeblin block on $S(H)$ with a strict margin and an RC1 modulus ensuring bounded-gap recurrence of that block on every tight minimizer in H. The Doeblin block yields a Birkhoff/Hilbert contraction factor $\tau < 1$ (Section 17) and bounded-gap recurrence converts this into exponential contraction with explicit (C, κ) (Theorem 17.7). The robustness layer converts the margins into an explicit radius ε^* with explicit degradation rules (Sections 25–28). Thus the returned artifact satisfies Definition 1.3.

This proves that DecideRUStable is a correct, witness-producing decision procedure for RU-Stable on the certified class. □

Theorem 1.5 (Exact frontier equivalence on the certified class)**.** *For a certified instance, the following statements are equivalent.*

- *The instance is* RU-Stable *(Definition 1.3).*
- *None of the obstruction predicates OG1–OG4 holds for the instance under the branch logic of Section 20.*
- DecideRUStable *returns a robust contraction certificate (and therefore does not return an obstruction witness).*

Equivalently, on the certified class,

$$\text{RU-Stable} \quad \Longleftrightarrow \quad \neg(\text{OG1} \vee \text{OG2} \vee \text{OG3} \vee \text{OG4}).$$

Proof. If DecideRUStable returns a robust contraction certificate, then by Definition 1.3 the instance is RU-Stable. This proves (3) $\Rightarrow$ (1).

If the instance is RU-Stable, then there exists a robust contraction certificate containing an aperiodic tight SCC, a valid face support, a Doeblin word with strict margin, and an RC1 recurrence modulus. Each of these fields negates the corresponding obstruction predicate OG1–OG4. Hence (1) $\Rightarrow$ (2).

Finally, if none of OG1–OG4 holds, then every obstruction branch is excluded and the only remaining branch of the finite logic is the contraction-certificate branch, so DecideRUStable returns a robust contraction certificate. Hence (2) $\Rightarrow$ (3). □

Corollary 1.6 (Maximal frontier report for DecideRUStable)**.** *In addition to returning a robust contraction certificate or an obstruction witness, the procedure can be required to output a* frontier report *(Section 21) whose* `minimal_gate` *is the first failing predicate in the branch logic and whose* `frontier_loci` *identify the implicated boundary loci. In the YES case, the frontier report records the decisive margins (notably γ and δ) together with the robust radius $\varepsilon^\star$. In the NO case, the frontier report points to the corresponding witness object and records the first obstruction type OG1–OG4.*

Theorem 1.7 (Unified certified local invariance radius for all outcomes)**.** *Fix a certified instance and run* DecideRUStable*. Let R denote the resulting frontier report (Corollary 1.6). Define the* decision invariance radius $\varepsilon_{\mathrm{dec}}$ *as follows.*

- *If R.*`minimal_gate` $=$ `contraction`*, set $\varepsilon_{\mathrm{dec}} := \varepsilon^\star$, the robust radius exported by the contraction certificate.*
- *If R.*`minimal_gate` $\in \{$`OG1`, `OG2`, `OG3`, `OG4`$\}$ *and the instance belongs to a fixed-support family in the sense of Definition 3.1, set*

  $$\varepsilon_{\mathrm{dec}} := \min\{\varepsilon_{\mathrm{tight}}, \varepsilon_{\mathrm{supp}}\},$$

  where $\varepsilon_{\mathrm{tight}}$ is the tight-core stability radius from Tight-Core Stability (Section 26) and $\varepsilon_{\mathrm{supp}}$ is the support-stability radius from Section 25.2. (If the perturbation model distinguishes costs and template entries, interpret the minimum componentwise.)

Then, for every perturbation of size at most $\varepsilon_{\mathrm{dec}}$ (in the stated perturbation model), the `minimal_gate` *label is invariant. In particular, the frontier report becomes a self-contained decision object that certifies not only the YES/NO outcome but also a nontrivial neighborhood on which that outcome is provably constant whenever the fixed-support contract applies.*

Proof. In the YES case, $\varepsilon^\star$ is chosen no larger than the tight-core and support-stability thresholds, and the robustness contract guarantees that the same witness word, recurrence modulus, and rate pack remain valid throughout the $\varepsilon^\star$-ball; hence the `minimal_gate` label remains `contraction`.

In the NO case under the fixed-support contract, tight-core stability preserves the tight SCC structure and the finite automata used by OG2–OG4, while support-stability preserves the Boolean support patterns on the relevant blocks. Therefore each obstruction predicate OG1–OG4 is invariant on the $\varepsilon_{\mathrm{dec}}$-ball (Theorem 29.1), and by ordered branch logic the first failing predicate (`minimal_gate`) is invariant as well. □

Corollary 1.8 (Quantitative frontier reports on both sides)**.** *On fixed-support families, every pipeline outcome admits a certified local invariance radius: $\varepsilon^\star$ in the contraction case, and $\min\{\varepsilon_{\mathrm{tight}}, \varepsilon_{\mathrm{supp}}\}$ in the obstruction cases. Consequently the decidability frontier is not merely a pointwise YES/NO classifier: it is a witness-producing classifier whose label is stable on an explicitly certified neighborhood whenever the model declares structural zeros.*

Remark 1.9 (Imprimitive tight cores and period blocking)**.** *The decision procedure is stated at step length 1 on the tight graph $G^{(0)}$. If OG1 occurs because every tight SCC is imprimitive, one may apply the finite p-step return-graph reduction described in Section 20 and rerun the same witness calculus on the blocked instance. Any contraction certificate on the blocked instance lifts to the original dynamics with rates rescaled by $1/p$.*

Remark 1.10 (Relation to JSR and switching stability)**.** *Theorem 1.4 does not claim a decision procedure for JSR in unrestricted classes. Instead, it identifies a concrete* decidable island*: within the present positive-template setting, one can decide whether a robust, quantitative uniform contraction mechanism exists, and one can locate precisely where this mechanism fails via explicit witnesses. This is consistent with, and complementary to, general undecidability barriers [1].*

1.5 Complexity bounds and bounded-language decision view

The decision procedures in this paper are finite and artifact-driven. This subsection records explicit worst-case bounds in terms of the sizes of the finite objects.

Proposition 1.11 (Size and runtime bounds for DECIDERUSTABLE)**.** *Fix a certified instance with tight SCC H and active face $S(H)$. Let $n := |V(H)|$, $m := |E(H)|$, and $s := |S(H)|$. Then the following bounds hold.*

- *Tight SCC decomposition and period computation run in $O(|V| + |E|)$ time on the tight graph.*
- *The Doeblin search (Algorithm 6) explores at most $n \cdot 2^{s^2}$ states, hence runs in $O\big(m \cdot 2^{s^2} \cdot s^3\big)$ time if Boolean products are implemented naively, and in $O\big(m \cdot 2^{s^2} \cdot s^2/\text{wordsize}\big)$ time with bitset operations.*
- *The avoidance-graph check (Algorithm 7) builds a graph with $n \cdot |w|$ vertices and at most $m \cdot |w|$ edges and decides RC1 in $O(m \cdot |w|)$ time.*
- *In canonical multi-block mode (Section 13), one builds an Aho–Corasick dictionary automaton on the canonical family $\mathcal{B}(H)$ with state count $|Q_{\mathrm{AC}}| = O\big(\sum_{b \in \mathcal{B}(H)} |b|\big)$ and then forms the lifted avoidance graph with $n \cdot |Q_{\mathrm{AC}}|$ vertices and at most $m \cdot |Q_{\mathrm{AC}}|$ edges; acyclicity (MB–RC1) is decided in $O(m \cdot |Q_{\mathrm{AC}}|)$ time.*

In particular, on fixed descriptor bounds where s is uniformly small, DECIDERUSTABLE *is effective with explicit finite search horizons and explicit witness sizes.*

Proof. The tight-core and SCC computations are standard linear-time graph routines. The Doeblin bound follows from the product-automaton construction in Section 12 and the state count in Proposition 12.1; each transition requires one Boolean product on $s \times s$ patterns. The avoidance-graph bounds follow from Definition 14.2 and the explicit vertex and edge counts used in Corollary 14.7. □

Corollary 1.12 (Bounded-language reduction to finite automata)**.** *Fix a tight SCC H and active face $S(H)$. Within the constrained language of tight edge words in H:*

- *existence of a Doeblin word is a reachability question in the finite product automaton of Section 12.2;*
- *the recurrence dichotomy for a witnessed word w is the emptiness/nonemptiness of the sofic avoidance shift $X_{\neg w}(H)$ (Proposition 14.5);*
- *consequently, the certificate-based stability decision is realized entirely by finite automata operating on the bounded language of the tight core.*

This provides an explicit instance of the general phenomenon that semigroup questions can become decidable under bounded-language restrictions [9].

1.6 Open-problem target registry and resolution map

Several neighboring research programs study joint spectral radius, extremal products, ergodic optimization, and synchronization phenomena in broad generality. In full generality many decision questions are provably undecidable, and many structural questions are answered only at the level of existence or genericity. The present work contributes a complementary kind of result: a *certificate calculus* that turns a range of these themes into finite objects whose validity is decidable on the certified positive-template class.

Target codes (OP registry). To keep claims mechanically checkable, we tag each open-problem family by a code of the form OP–*Theme*–*Target*. A code refers to a precise statement *within the certified class* together with the pipeline artifact that realizes it. The remainder of this section records the principal codes currently supported by the paper.

The table below records, for each commonly cited open-problem family, the concrete statement resolved *within the certified class* and the pipeline artifact that realizes the statement.

Interpretation. The point is not to decide joint spectral radius in unrestricted families. Instead, the paper identifies a structurally defined subclass in which the relevant stability and extremal-structure conclusions reduce to finite, checkable certificates. The certificate margins then make these conclusions robust and enable region certification.

OP–JSR–RU (decidability frontier). Theorem 1.4 gives a dichotomy. Theorem 1.5 sharpens this to an explicit iff on the certified class. Either a robust certificate exists, producing explicit contraction rates and a robustness radius ε^* (Definition 1.3), or an obstruction witness exists (OG1–OG4), and the witness is finite and verifiable. The frontier report can be checked independently via Algorithm 8. This realizes a decidable island inside the broader JSR landscape where global decision problems are known to be hard and can be undecidable [1].

OP–MATHER–EFF (effective extremal language). Within the certified class, the boundary carrier $\Sigma_{\mathcal{A}}$ is a computable SFT, and the Doeblin/recurrence layer refines the model to a verifiable sofic package that records either bounded-gap recurrence (RC1) or an explicit avoidance cycle (OG4). The resulting export object `MatherSoficModel` (Section 24) is a compact finite-state summary of the extremal itinerary structure, and it serves as the effective analogue of the qualitative Mather-set viewpoint in the ergodic-optimization literature [2].

OP–CHAMBER–STAB and OP–BOUNDARY–LOC (chambers and failure loci). The chamber machinery (Section 35) states that, away from explicit tie and margin-collapse loci, the tight combinatorics and the obstruction regime are locally constant. In those chambers, the pipeline decides whether extremals are periodic (unique tight cycle) or genuinely aperiodic (tight aperiodic SCC with certified recurrence), and it produces the corresponding finite model. The region certification layer (Section 30) localizes where chamber stability can fail by explicit margin collapse.

OP–SYNC–IND, OP–PRIM–LEN, OP–EXTNORM–CONS, OP–BLANG–DEC. The remaining codes record how induced automata, Doeblin-length bounds, constructive extremal geometry, and bounded-language decidability phenomena are realized by explicit pipeline artifacts within the certified class.

OP–FRONTIER–MAP (operational frontier mapping). The obstruction taxonomy (OG1–OG4) and the robustness margins provide an operational map of why certification fails, returning concrete finite witnesses rather than leaving failure as an opaque non-result. The region-level completion of the same idea is the `FrontierAtlasReport` (Proposition 31.8 and Theorem 22.3), which we treat as the default exported object for OP–FRONTIER–MAP and OP–ATLAS–FIN; it exports the chamberwise minimal-gate stratification, a carry-forward decision radius for each chamber, and an explicit localization of any uncovered remainder inside declared degeneracy loci.

Toward cover/atlas universality on compact normalized families within the certified class. Theorem 31.2 isolates the remaining step needed to upgrade from local robustness to a global statement on a compact region Θ: prove that the named margin-collapse loci are absent on Θ (equivalently: obtain uniform positive lower bounds for the certificate margins). This can be achieved either by structural hypotheses that enforce uniform gaps, or by region mode plus a finite covering argument that certifies Θ and produces a global robustness radius. If Θ intersects the named margin-collapse loci, the global procedure still terminates with a boundary-atlas report that localizes the degeneracy/obstruction region rather than asserting uniform contraction.

2 Canonical Normalization and the Tight Subgraph

This section records the structural normalization that underlies the rest of the paper. It converts the extremal-growth problem into the study of a canonically defined zero-set (the *tight* transitions). Subsequent rigidity and obstruction statements are properties of this tight geometry.

2.1 Reduced costs

Let $G = (V, E)$ be the reduced finite graph produced by the descriptor reduction, with edge costs $c : E \to \mathbb{R}$. Define the minimal cycle mean

$$\lambda_* := \min_{\gamma \text{ cycle}} \frac{1}{|\gamma|} \sum_{e \in \gamma} c(e).$$

Theorem 2.1 (Canonical normalization by a potential)**.** *There exists a function (potential) $\phi : V \to \mathbb{R}$ such that the* reduced costs

$$c_\phi(e) := c(e) - \lambda_* + \phi(\mathrm{tail}(e)) - \phi(\mathrm{head}(e))$$

satisfy $c_\phi(e) \geq 0$ for all $e \in E$.

Proof. Let $\tilde{c}(e) := c(e) - \lambda_*$. For any directed cycle γ of length k, $\sum_{e\in\gamma} \tilde{c}(e) = k(\bar{c}(\gamma) - \lambda_*) \geq 0$, so $(G, \tilde{c})$ has no negative cycles. Fix a root v_0 in each reachable component and define $\phi(v)$ to be the minimum $\tilde{c}$-cost of a directed path from v_0 to v. Then for any edge $e : u \to v$, optimality of $\phi(v)$ yields $\phi(v) \leq \phi(u) + \tilde{c}(e)$, i.e. $\tilde{c}(e) + \phi(u) - \phi(v) \geq 0$. This inequality is exactly $c_\phi(e) \geq 0$. □

2.2 The tight subgraph

Definition 2.2 (Tight edges and tight subgraph)**.** *Fix a potential ϕ as in Theorem 2.1. Define the tight edges*

$$E^{(0)} := \{e \in E : \ c_\phi(e) = 0\}, \qquad G^{(0)} := (V^{(0)}, E^{(0)}),$$

where $V^{(0)}$ are vertices incident to $E^{(0)}$.

2.3 Period and aperiodicity of a tight SCC

Definition 2.3 (Period of a strongly connected digraph)**.** *Let H be a strongly connected directed graph. Its* period *is*

$$\mathrm{per}(H) := \gcd\{\, |\gamma| : \gamma \textit{ is a directed cycle in } H \,\}.$$

We call H aperiodic *if* $\mathrm{per}(H) = 1$*, and* imprimitive *otherwise.*

2.4 Minimizers are asymptotically tight

A (one-sided) path $\pi = (e_0, e_1, \dots)$ has length-n average cost

$$\mathrm{Avg}_n(\pi) := \frac{1}{n}\sum_{t=0}^{n-1} c(e_t).$$

We call π *minimizing* if $\liminf_{n\to\infty} \mathrm{Avg}_n(\pi) = \lambda_*$.

Proposition 2.4 (Asymptotic tightness of minimizing paths)**.** *Let π be a minimizing path and fix ϕ as above. Then the set of times t with $e_t \notin E^{(0)}$ has asymptotic density 0. Equivalently, $E^{(0)}$ is visited with asymptotic density 1 along π.*

Proof. Because E is finite and $c_\phi(e) \geq 0$, define

$$\alpha := \min\{c_\phi(e) : e \in E \setminus E^{(0)}\},$$

with the convention $\alpha = +\infty$ if $E^{(0)} = E$. When $E^{(0)} \neq E$, we have $\alpha > 0$. Let N_n be the number of indices $t < n$ with $e_t \notin E^{(0)}$. Then $\sum_{t=0}^{n-1} c_\phi(e_t) \geq \alpha N_n$. On the other hand, telescoping the potential gives

$$\sum_{t=0}^{n-1} c_\phi(e_t) = \sum_{t=0}^{n-1} \big(c(e_t) - \lambda_*\big) + \phi(v_0) - \phi(v_n),$$

where v_n is the vertex reached after n steps. Since ϕ is bounded on the finite reachable vertex set, $|\phi(v_0) - \phi(v_n)| \leq C$. Divide by n and take $\liminf$. Minimizing implies $\liminf_{n\to\infty} \frac{1}{n}\sum_{t=0}^{n-1}(c(e_t) - \lambda_*) = 0$, hence $\liminf_{n\to\infty} \frac{1}{n}\sum_{t=0}^{n-1} c_\phi(e_t) = 0$. Therefore $\liminf_{n\to\infty} \alpha N_n/n = 0$, which forces $N_n/n \to 0$. $\square$

2.5 A tight-core dichotomy

For a subset $A \subseteq \mathbb{N}$, write its *lower asymptotic density* as

$$\underline{d}(A) := \liminf_{n\to\infty} \frac{|A \cap \{0, 1, \dots, n-1\}|}{n}.$$

Proposition 2.5 (Tight-core dichotomy)**.** *Let π be a minimizing path and let $T := \{t \geq 0 :\ e_t \in E^{(0)}\}$ be its set of tight times. For each SCC H of the tight graph $G^{(0)}$, set*

$$T_H := \{t \in T :\ v_t \in H\},$$

where v_t is the tail vertex of e_t. Then there exists at least one SCC H with $\underline{d}(T_H) > 0$. Moreover, exactly one of the following holds:

- *(P) every SCC H with $\underline{d}(T_H) > 0$ is imprimitive, i.e.* $\mathrm{per}(H) > 1$ *(Definition 2.3);*
- *(A) there exists an aperiodic SCC H of $G^{(0)}$ with $\underline{d}(T_H) > 0$, i.e.* $\mathrm{per}(H) = 1$.

Proof. Let $T := \{t \geq 0 :\ e_t \in E^{(0)}\}$ be the set of *tight times* along π. By Proposition 2.4, T has asymptotic density 1, hence lower density 1. For each strongly connected component H of the tight graph $G^{(0)}$, set

$$T_H := \{t \in T :\ \text{the time-}t\text{ vertex } v_t \text{ lies in } H\}.$$

The sets $(T_H)_H$ form a *finite* partition of T indexed by the SCCs of $G^{(0)}$. If every T_H had lower density 0, then their finite union T would also have lower density 0, contradicting $\underline{d}(T) = 1$. Therefore there exists at least one SCC H with $\underline{d}(T_H) > 0$.

If among the SCCs with $\underline{d}(T_H) > 0$ there is an aperiodic one, we are in case (A). Otherwise every SCC visited with positive lower density along tight times is imprimitive, which is case (P). □

Remark 2.6. *The remainder of the paper analyzes the geometry of $G^{(0)}$. Rigidity (projective contraction) is obtained from recurrence and Doeblin structure inside an aperiodic tight SCC. Obstructions are finite failures of the corresponding tight-core regularity conditions.*

3 A concrete compact normalized target family Θ_{can} (ready-to-run)

The global machinery of Sections 31–32 applies to any compact normalized parameter family. To remove ambiguity about what "choose Θ" means in practice, we record a canonical, concrete choice that is broad enough to cover most finite-template models while remaining fully compatible with the robust-certificate contract.

3.1 Raw parameters and normalization

Fix once and for all:

- a finite reduced descriptor digraph $G = (V, E)$ together with a fixed labeling map $e \mapsto T_e$ into nonnegative $d \times d$ templates,
- the *support patterns* of all templates T_e (which entries are structurally allowed to be positive),
- and the perturbation metrics used in Robust Certificate Contract (entrywise for templates and a norm on costs).

3.2 Support-pattern contract (structural zeros are part of the model)

Most finite-template models do not allow arbitrary new interactions to appear under perturbation: the *sparsity patterns* of the templates are part of the model definition (legal vs. illegal transitions, prohibited couplings, forbidden edges, etc.). Accordingly, the certified class in this paper treats zeros as *structural*.

Definition 3.1 (Fixed-support template-family contract)**.** *A parameter family Θ satisfies the* fixed-support contract *if for each labeled template T_e there is a fixed set of allowed indices*

$$\Sigma_e \subseteq \{1, \dots, d\} \times \{1, \dots, d\}$$

such that for every $\theta \in \Theta$,

$$(T_e(\theta))_{ij} > 0 \iff (i,j) \in \Sigma_e, \qquad (T_e(\theta))_{ij} = 0 \iff (i,j) \notin \Sigma_e.$$

Equivalently, perturbations within Θ are support-preserving in the sense of Definition 25.1.

Lemma 3.2 (Uniform support-stability on Θ_{can})**.** *On $\Theta_{\mathrm{can}}(m_0, C_0)$ every allowed template entry lies in $[m_0, 1]$ after normalization. Consequently the support margin η of Definition 25.3 satisfies $\eta \geq m_0$, and for absolute entrywise perturbations one may take*

$$\varepsilon_{\mathrm{supp}} \geq m_0/2.$$

Proof. By definition of Θ_{can}, every structurally allowed entry satisfies $(T_e(\theta))_{ij} \in [m_0, 1]$, hence the minimum over positive entries is at least m_0. The claimed bound on $\varepsilon_{\mathrm{supp}}$ follows from Definition 25.3. □

A raw parameter θ consists of:

- **template entry weights:** for each structurally allowed entry (i, j) of each label T_e, a positive real value $a_{e,ij}$,
- **edge costs:** a real cost vector $c_\theta : E \to \mathbb{R}$ (interpreted as reduced costs after the canonical normalization of Section 2).

To eliminate noncompact scaling degrees of freedom, we use the entrywise-max normalization: for each label e, rescale the nonzero entries of $T_e(\theta)$ so that

$$\max_{i,j}(T_e(\theta))_{ij} = 1.$$

This normalization is continuous on the strictly positive locus and is compatible with the projective nature of the contraction conclusions.

3.3 The canonical compact family

Fix constants

$$0 < m_0 \leq 1, \qquad C_0 > 0.$$

Define $\Theta_{\mathrm{can}} = \Theta_{\mathrm{can}}(m_0, C_0)$ to be the set of all normalized parameters θ such that:

- every structurally allowed entry of every template satisfies
$$(T_e(\theta))_{ij} \in [m_0, 1],$$
- the costs satisfy a mean-zero normalization and a uniform box bound
$$\sum_{e \in E} c_\theta(e) = 0, \qquad |c_\theta(e)| \leq C_0 \ \text{ for all } e \in E.$$

Lemma 3.3 (Compactness of Θ_{can})**.** *Θ_{can} is compact in the product topology induced by the Robust Certificate Contract norms.*

Proof. Each template-entry coordinate lies in the compact interval $[m_0, 1]$ and the cost coordinates lie in $[-C_0, C_0]$ with one closed linear constraint $\sum c = 0$. Finite products of compact sets are compact, and closed subsets of compact sets are compact. □

3.4 What "completion" means on Θ_{can}

On Θ_{can} the global decider of Theorem 32.3 is now a concrete, fully specified procedure: the input is the machine description of Θ_{can} (box constraints plus normalization), and the output is a `GlobalUniversalityReport`.

The only remaining mathematical distinction is *which outcome occurs*:

- If Θ_{can} avoids the degeneracy loci (Definition 31.3) chamberwise, then the decider must certify (Corollary 32.5) and yields *good-global contraction universality within the certified class on the compact normalized family* Θ_{can}.
- Otherwise, the decider returns a *boundary-global regime atlas* locating the obstruction/degeneracy neighborhood where uniform contraction cannot hold on all of Θ_{can} without further restriction.

3.5 Making the net-certification inputs explicit (no hidden regularity)

The "global-on-Θ by finite net" route in Proposition 31.14 requires conservative continuity moduli. To prevent any impression that these are being assumed implicitly, the parameter-family descriptor schema includes a dedicated contract block `net_certification_contract`.

Concretely, when one wants a certified uniform margin lower bound via Lemma 31.13, the descriptor must declare:

- the parameter norm used to build an h-net;
- a conservative Lipschitz constant L_c for $\theta \mapsto c(\theta)$ in $\|\cdot\|_\infty$;
- a conservative Lipschitz constant L_A for $\theta \mapsto A^{(k)}(\theta)$ in entrywise $\|\cdot\|_{\infty,\mathrm{ent}}$;
- a uniform entry bound $M \geq \sup_{\theta\in\Theta} \max_{k,i,j} A_{ij}^{(k)}(\theta)$.

On a frozen witness chamber, these inputs yield explicit envelopes for L_γ and L_δ via Lemmas 31.18 and 31.19.

Remark 3.4 (Concrete machine description files)**.**

4 Aubry–Tight Core and the Induced Boundary Subshift

This section pins down the precise meaning of *core* used throughout the paper. The definition is finite, canonical, and built from the normalized reduced costs. It isolates the locally optimal transitions and the induced symbolic system that supports calibrated minimizers.

4.1 Calibrated subactions and local optimality

Fix the reduced graph $G = (V, E)$ with costs $c : E \to \mathbb{R}$ and the extremal mean λ_*.

Definition 4.1 (Calibrated subaction)**.** *A function $u : V \to \mathbb{R}$ is a* calibrated subaction *(for c at level λ_*) if for every edge $e \in E$,*

$$u(\mathrm{head}(e)) \;\leq\; u(\mathrm{tail}(e)) + c(e) - \lambda_*. \tag{1}$$

Equivalently, the reduced cost

$$c_u(e) := c(e) - \lambda_* + u(\mathrm{tail}(e)) - u(\mathrm{head}(e)) \tag{2}$$

satisfies $c_u(e) \geq 0$ for all $e \in E$.

Remark 4.2. *The potential ϕ produced by Theorem 2.1 is a calibrated subaction in the sense of Definition 4.1, with $c_\phi = c_u$. Conversely, any calibrated subaction yields a valid canonical normalization.*

Lemma 4.3 (Tight edges are exactly locally optimal edges)**.** *Let u be any calibrated subaction. Then an edge $e \in E$ is tight (i.e. $c_u(e) = 0$) if and only if equality holds in (1):*

$$u(\mathrm{head}(e)) = u(\mathrm{tail}(e)) + c(e) - \lambda_*.$$

In particular, the tight edge set $E^{(0)} = \{e : c_u(e) = 0\}$ is exactly the set of one-step transitions that are locally optimal with respect to the calibrated subaction.

Proof. By definition, $c_u(e) = c(e) - \lambda_* + u(\mathrm{tail}(e)) - u(\mathrm{head}(e))$. The inequality (1) is equivalent to $c_u(e) \geq 0$. Thus $c_u(e) = 0$ holds exactly when (1) holds with equality. □

4.2 Aubry–tight core

Fix a calibrated subaction u (for example $u = \phi$). Let $G^{(0)} = (V^{(0)}, E^{(0)})$ denote the tight subgraph, i.e. the subgraph of locally optimal edges.

Definition 4.4 (Aubry–tight core)**.** *The* Aubry–tight core *(or simply* the core*) is the union of the strongly connected components of $G^{(0)}$ that contain at least one directed cycle. Equivalently, it is the set of vertices that lie on some tight directed cycle. We denote this vertex set by $\mathcal{A} \subseteq V^{(0)}$ and write $G^{\mathcal{A}}$ for the induced subgraph of $G^{(0)}$ on $\mathcal{A}$.*

Remark 4.5. *Throughout the paper, when we refer to a* tight core*,* core SCC*, or* aperiodic core*, we mean an SCC of $G^{\mathcal{A}}$ as in Definition 4.4. No statement later relies on any stronger "eventual confinement" property for arbitrary mean-minimizers.*

4.3 Induced boundary subshift

The tight core defines a finite-type symbolic system capturing calibrated two-sided minimizers.

Definition 4.6 (Boundary subshift induced by the tight core)**.** *Let $\Sigma_{\mathcal{A}} \subset (E^{(0)})^{\mathbb{Z}}$ be the set of bi-infinite edge sequences $\omega = (\dots, e_{-1}, e_0, e_1, \dots)$ such that consecutive edges match and every vertex visited lies in $\mathcal{A}$. Equivalently, $\Sigma_{\mathcal{A}}$ is the two-sided edge shift of the finite directed graph $G^{\mathcal{A}}$. See, e.g., Lind and Marcus [17] for standard background on shifts of finite type and edge shifts. We call $(\Sigma_{\mathcal{A}}, \sigma)$ the* boundary subshift *induced by the Aubry–tight core.*

Lemma 4.7 (Tight paths have extremal mean)**.** *Let $(e_0, \dots, e_{n-1})$ be a length-n path consisting only of tight edges in $E^{(0)}$. Then*

$$\frac{1}{n}\sum_{t=0}^{n-1} c(e_t) = \lambda_* + \frac{u(v_n) - u(v_0)}{n},$$

where v_0 is the initial vertex, v_n is the terminal vertex, and u is any calibrated subaction. In particular, along any one-sided infinite tight path, the asymptotic average cost equals λ_*.

Proof. For each tight edge $e_t : v_t \to v_{t+1}$ we have $c_u(e_t) = 0$, i.e. $c(e_t) - \lambda_* + u(v_t) - u(v_{t+1}) = 0$. Summing from $t = 0$ to $n-1$ yields $\sum_{t=0}^{n-1} c(e_t) = n\lambda_* + u(v_n) - u(v_0)$. Divide by n. □

Remark 4.8. *The boundary subshift $\Sigma_{\mathcal{A}}$ is the natural symbolic carrier of calibrated (two-sided) minimizers. The gap-free boundary classification later in the paper is performed SCC-by-SCC on this core subshift. When working with a specific one-sided minimizing trajectory, the relevant input is the set of* tight times *(Proposition 2.4) and the positive-density core selection (Proposition 2.5); no stronger trapping statement is assumed.*

5 Computing the extremal mean lambda-star, a calibrating potential, and the tight subgraph

This appendix records an explicit finite procedure to compute the objects in Section 2. All steps are explicit finite graph algorithms; we include them to make the normalization fully machine-checkable.

5.1 Computing the minimal cycle mean

Let $G = (V, E)$ with $|V| = n$, $|E| = m$, and costs $c : E \to \mathbb{R}$. The minimal cycle mean

$$\lambda_* = \min_{\gamma \text{ cycle}} \frac{1}{|\gamma|} \sum_{e \in \gamma} c(e)$$

can be computed in $O(nm)$ time using Karp's algorithm.

Algorithm 1 Karp minimal cycle mean

Require: Directed graph $G = (V, E)$, costs $c(e)$
Ensure: λ_*

```
Choose any ordering of vertices and initialize d_0(v) = 0 for all v ∈ V
for k = 1 to n do
    for all v ∈ V do
        d_k(v) ← min_{(u→v)∈E} (d_{k−1}(u) + c(u → v))
    end for
end for
for all v ∈ V do
    M(v) ← max_{0≤k≤n−1} (d_n(v) − d_k(v))/(n − k)
end for
λ_* ← min_{v∈V} M(v)
return λ_*
```

5.2 Computing a calibrating potential

Set $\tilde{c}(e) = c(e) - \lambda_*$. Then G has no negative cycles with respect to $\tilde{c}$ by definition of λ_*. A potential ϕ such that $c_\phi(e) = \tilde{c}(e) + \phi(u) - \phi(v) \geq 0$ for all $e : u \to v$ can be obtained from shortest-path distances.

Algorithm 2 Calibrating potential via shortest paths

Require: Graph $G = (V, E)$, reduced costs $\tilde{c}(e) = c(e) - \lambda_*$
Ensure: Potential ϕ with $c_\phi(e) \geq 0$ for all e

```
Add a super-source s with zero-cost edges (s → v) to every v ∈ V
Run Bellman–Ford on (V ∪ {s}, E ∪ {(s → v)}) with costs c̃ to compute distances φ(v) =
    dist(s, v)
return φ
```

Remark 5.1. *Because there are no negative cycles, Bellman–Ford terminates and returns finite distances. The inequality $\phi(v) \leq \phi(u) + \tilde{c}(u \to v)$ is exactly $c_\phi(u \to v) \geq 0$.*

5.3 Extracting the tight subgraph and its SCC structure

Given ϕ, compute $c_\phi(e)$ and define $E^{(0)} = \{e : c_\phi(e) = 0\}$ as in Definition 2.2. This yields the tight subgraph $G^{(0)}$. SCC decomposition runs in $O(n + m)$ time (Tarjan/Kosaraju). Aperiodicity of an SCC is certified by computing the gcd of its cycle lengths (linear-time in the SCC via a spanning tree + back-edge depth differences).

Algorithm 3 Tight subgraph + SCC periodicity labels

Require: $G = (V, E)$, costs c, λ_*, potential ϕ
Ensure: Tight graph $G^{(0)}$ and SCCs labeled periodic/aperiodic
$E^{(0)} \leftarrow \{u \to v \in E : \; c(u \to v) - \lambda_* + \phi(u) - \phi(v) = 0\}$
Compute SCC decomposition of $G^{(0)}$
for all SCCs H **do**
  Compute $\mathrm{per}(H) = \gcd$ of cycle lengths in H
  if $\mathrm{per}(H) = 1$ and $|H| > 1$ **then** label aperiodic
  elselabel periodic
  end if
end for

5.4 Complexity summary

All objects in Section 2 are computable in polynomial time from the reduced finite graph:

- Karp minimal mean: $O(nm)$.
- Bellman–Ford potential: $O(nm)$.
- Tight graph extraction: $O(m)$.
- SCC decomposition: $O(n + m)$.

Thus the canonical normalization and tight geometry can be produced and checked algorithmically as part of the finite certification pipeline.

6 Calibrated Minimizers and Selection Principles

This section isolates the precise sense in which one may *restrict* attention to calibrated (locally optimal) minimizers, while keeping clear what this does *not* imply for arbitrary one-sided mean-minimizers. The main point is that calibration is an *edgewise* (one-step) equality condition, so calibrated minimizers live entirely on the tight subgraph, whereas general minimizers are only asymptotically tight.

6.1 Calibrated minimizers

Fix a calibrated subaction $u : V \to \mathbb{R}$ in the sense of Definition 4.1, and write c_u for the associated reduced costs (2).

Definition 6.1 (Calibrated bi-infinite minimizer)**.** *A bi-infinite admissible edge sequence $\omega = (\dots, e_{-1}, e_0, e_1, \dots) \in E^{\mathbb{Z}}$ is a* calibrated minimizer for u *if consecutive edges match and*

$$c_u(e_t) = 0 \quad \textit{for all } t \in \mathbb{Z}. \tag{3}$$

Equivalently, ω is a bi-infinite path in the tight subgraph $G^{(0)} = (V^{(0)}, E^{(0)})$ defined by u.

Lemma 6.2 (Calibrated minimizers are supported on tight edges)**.** *If ω is a calibrated minimizer for u, then every edge of ω lies in $E^{(0)} = \{e \in E : c_u(e) = 0\}$. In particular, every calibrated minimizer is minimizing with asymptotic mean cost λ_*.*

Proof. The first claim is immediate from (3). For the mean cost, sum the identity $c(e_t) - \lambda_* + u(v_t) - u(v_{t+1}) = 0$ along any finite window, exactly as in Lemma 4.7, and divide by the window length. □

6.2 Existence on the Aubry–tight core

The Aubry–tight core $\mathcal{A}$ from Definition 4.4 is designed so that tight cycles exist through every vertex of $\mathcal{A}$. This yields calibrated minimizers without any recurrence hypothesis.

Proposition 6.3 (Periodic calibrated minimizers on the core). *For every vertex $v \in \mathcal{A}$ there exists a periodic bi-infinite calibrated minimizer $\omega \in \Sigma_{\mathcal{A}}$ that visits v. Consequently, $\Sigma_{\mathcal{A}}$ is nonempty whenever $\mathcal{A} \neq \varnothing$.*

Proof. By Definition 4.4, the vertex v lies on a directed tight cycle γ in $G^{(0)}$. Let w be the finite edge word given by γ starting and ending at v. Repeating w periodically in both time directions produces a bi-infinite admissible edge sequence $\omega \in (E^{(0)})^{\mathbb{Z}}$. Then $c_u(e_t) = 0$ for all t, so ω is calibrated by Definition 6.1. □

6.3 Minimizing measures are tight

The previous subsection constructs explicit calibrated minimizers on $\mathcal{A}$. For applications, it is also useful to know that every *invariant* minimizer is supported on the tight shift.

Proposition 6.4 (Invariant minimizers live on the tight shift). *Let μ be a σ-invariant probability measure on the two-sided edge shift of G. If μ minimizes the average cost, i.e.*

$$\int c\, d\mu = \lambda_*,$$

then μ is supported on the tight edge shift $(E^{(0)})^{\mathbb{Z}}$. In particular, $c_u(e) = 0$ for μ-almost every edge occurrence.

Proof. By Definition 4.1, the reduced cost $c_u(e) \geq 0$ for all edges. For a bi-infinite path ω with vertex sequence $(v_t)_{t\in\mathbb{Z}}$ we have

$$c_u(e_t) = c(e_t) - \lambda_* + u(v_t) - u(v_{t+1}).$$

Integrating with respect to a σ-invariant measure μ yields

$$\int c_u\, d\mu = \int (c - \lambda_*)\, d\mu + \int (u \circ \mathrm{tail} - u \circ \mathrm{head})\, d\mu.$$

The last integral vanishes by σ-invariance (it is the integral of a coboundary), hence

$$\int c_u\, d\mu = \int c\, d\mu - \lambda_*.$$

If $\int c\, d\mu = \lambda_*$, then $\int c_u\, d\mu = 0$. Since $c_u \geq 0$, it follows that $c_u = 0$ μ-almost surely. Equivalently, μ is supported on the set of paths using only edges in $E^{(0)}$. □

Corollary 6.5 (Calibrated representatives exist in minimizing supports). *If μ is a σ-invariant minimizing measure, then μ-almost every ω is a calibrated minimizer for u. In particular, the support of any minimizing measure is contained in the boundary subshift $\Sigma_{\mathcal{A}}$.*

Proof. The first statement is immediate from Proposition 6.4. For the second, note that any bi-infinite tight path must lie in an SCC of $G^{(0)}$ that contains a cycle. By Definition 4.4, the union of such SCCs is exactly the Aubry–tight core $\mathcal{A}$, so every bi-infinite tight path lies in $\Sigma_{\mathcal{A}}$. □

6.4 What calibration does and does not buy for one-sided minimizers

The results above justify a common proof strategy: when proving statements about the *symbolic carrier* of extremality (the boundary subshift and its SCCs), one may work entirely on calibrated minimizers. However, one must not confuse this with a statement about a *given* one-sided minimizing trajectory.

Proposition 6.6 (Calibrated selection from a one-sided minimizer)**.** *Let $\pi = (e_0, e_1, \dots)$ be a one-sided minimizing path. Then there exists a bi-infinite calibrated minimizer ω and a sequence of times $n_k \to \infty$ such that $\sigma^{n_k}\pi \to \omega$ in the product topology on $E^{\mathbb{Z}}$. Equivalently, the ω-limit set of π contains a calibrated minimizer.*

Proof. By Proposition 2.4, the set $T = \{t \ge 0 : e_t \in E^{(0)}\}$ has asymptotic density 1. Fix an integer $m \ge 1$. Because T has density 1, there exist arbitrarily large times n such that the window $\{n - m, \dots, n + m\}$ is contained in T. Choose $n(m)$ with this property. The shift space $E^{\mathbb{Z}}$ is compact, so by a diagonal subsequence argument there exists a sequence $n_k \to \infty$ with $\sigma^{n_k}\pi$ converging to some bi-infinite admissible limit path ω. By construction of the windows, for each fixed coordinate $t \in \mathbb{Z}$, the edge at position t in $\sigma^{n_k}\pi$ belongs to $E^{(0)}$ for all sufficiently large k; hence the limiting edge $e_t(\omega)$ also lies in $E^{(0)}$. Therefore ω uses only tight edges and is a calibrated minimizer by Definition 6.1. □

Remark 6.7 (Scope and limitations)**.** *Proposition 6.6 is a* selection *statement: from a minimizing trajectory one can extract calibrated minimizers as limit points. It does not imply that the original trajectory is eventually tight, trapped in a single tight SCC, or recurrent. The only trajectory-level facts used later are*

- *density-*1 *tightness (Proposition 2.4), and*
- *positive-lower-density selection of at least one tight SCC (Proposition 2.5).*

Any stronger recurrence input (bounded gaps of a certified Doeblin word, or a syndetic modulus from an automaton) is treated as a separate layer.

7 Minimality Route: what can and cannot be forced on the tight core

This section clarifies the "minimality" upgrade route for the minimizing dynamics on the tight core. The conclusion is sharp and finite-witness: because the boundary carrier $\Sigma_{\mathcal{A}}$ is a shift of finite type, true topological minimality (and hence linear recurrence) occurs only in the purely periodic case. Accordingly, minimality is not an automatic consequence of calibration or tightness; the correct replacement for gap-control is the *finite* recurrence certificate layer (RC1) and the slack-forcing route developed later.

7.1 Minimal SFTs are periodic

We record the elementary structural fact that blocks the naive minimality upgrade on SFT carriers.

Definition 7.1 (Minimality)**.** *A two-sided subshift (X, σ) is* minimal *if every orbit is dense in X. Equivalently, X has no proper nonempty closed σ-invariant subsets.*

Lemma 7.2 (A minimal edge shift is a single cycle)**.** *Let H be a finite directed graph and let X_H be its two-sided edge shift. If (X_H, σ) is minimal, then X_H consists of a single periodic orbit. Equivalently, H is a directed cycle (every vertex has in-degree and out-degree equal to 1 in the active component).*

Proof. If X_H is nonempty, then H contains a directed cycle, hence X_H contains a periodic point ω obtained by repeating that cycle. The orbit closure of a periodic point is finite. If X_H were minimal, the orbit closure of ω would have to equal all of X_H. Therefore X_H is finite. An edge shift over a finite graph is finite if and only if the underlying active component is a directed cycle, because any branching (a vertex with two distinct outgoing edges or two distinct incoming edges) produces at least two distinct infinite paths. Thus H must be a directed cycle and X_H is a single periodic orbit. □

7.2 Implication for the minimizing carrier

Recall that the boundary carrier $\Sigma_{\mathcal{A}}$ defined in Definition 4.6 is the two-sided edge shift of the finite graph $G^{\mathcal{A}}$. Hence it is a shift of finite type.

Proposition 7.3 (Minimality on $\Sigma_{\mathcal{A}}$ is a periodic obstruction)**.** *If the boundary carrier $(\Sigma_{\mathcal{A}}, \sigma)$ is minimal, then every SCC of $G^{\mathcal{A}}$ is a directed cycle. In particular, the only minimal regime for the tight-core carrier is the purely periodic regime.*

Conversely, if a tight SCC $H \subseteq G^{\mathcal{A}}$ is a directed cycle, then its edge shift is minimal and every tight trajectory supported in H is periodic.

Proof. Apply Lemma 7.2 to each nonempty irreducible component. If $\Sigma_{\mathcal{A}}$ is minimal and nonempty, it cannot decompose into multiple invariant components, so it coincides with the shift of a single SCC. That SCC must be a directed cycle, and the conclusions follow. The converse direction is immediate for a directed cycle. □

7.3 The exact missing property and the correct replacement

The previous proposition isolates the precise missing property: without an explicit *finite* obstruction witness forcing the tight SCC to be a directed cycle, minimality (and linear recurrence) cannot be deduced from calibration alone. In particular, tightness is compatible with:

- multiple tight cycles in the same tight SCC,
- branching tight choices (out-degree ≥ 2 along tight edges),
- aperiodic tight SCCs that are topologically transitive but far from minimal.

These shapes are not pathologies; they occur naturally and are consistent with the obstruction taxonomy OG1–OG3. Therefore, the minimality route is not the right vehicle for gap-free recurrence on minimizers.

Remark 7.4 (Finite recurrence certificate as the correct "minimality substitute")**.** *What is actually used in the contraction upgrade is not minimality of the entire tight SCC, but a finite return-control statement for a* specific certified Doeblin word w*. This is exactly the role of the recurrence certificate layer (RC1): for a fixed tight SCC H and a candidate word w supported in H, one either*

- *produces an explicit bounded-gap modulus L_{syn} guaranteeing syndetic returns of w along* every *infinite tight path in H, or*

- *returns a finite avoidance-cycle witness showing that some periodic tight path avoids w entirely.*

This finite dichotomy is the sharp replacement for minimality in the gap-free regime.

8 Slack-Forcing from Doeblin-Word Avoidance

This section isolates a quantitative principle that will be used later in the forcing dichotomy: once a Doeblin word is certified on a tight aperiodic SCC and the bounded-gap recurrence certificate (RC1) holds for tight trajectories, then any attempt to avoid that word over long time windows necessarily incurs a *linear* amount of reduced-cost slack.

8.1 Setup and constants

Fix the calibrated potential ϕ and reduced costs $c_\phi(e) \geq 0$ as in (2). Recall the tight edge set $E^{(0)} = \{e : c_\phi(e) = 0\}$ and the tight graph $G^{(0)} = (V^{(0)}, E^{(0)})$.

Let

$$\alpha := \min\{c_\phi(e) : e \in E \setminus E^{(0)}\}, \tag{4}$$

with the convention $\alpha = +\infty$ if $E^{(0)} = E$. When $E^{(0)} \neq E$, we have $\alpha > 0$ by finiteness of E. This constant already appears in Proposition 2.4.

Definition 8.1 (Slack count and slack mass on a window)**.** *For a finite edge-word $\gamma = (e_0, \dots, e_{L-1})$, define*

$$\mathrm{SlackCount}(\gamma) := \left|\{0 \leq t < L : \ e_t \notin E^{(0)}\}\right|, \qquad \mathrm{SlackMass}(\gamma) := \sum_{t=0}^{L-1} c_\phi(e_t).$$

By construction, $\mathrm{SlackMass}(\gamma) \geq \alpha \cdot \mathrm{SlackCount}(\gamma)$ whenever $\alpha < \infty$.

8.2 RC1 turns word-avoidance into a bound on tight-run lengths

Let H be a tight SCC of $G^{(0)}$ and let w be a finite tight edge-word supported in H. Assume that the recurrence certificate **(RC1)** holds for (H, w) in the sense of Definition 14.1. Let $L_{\mathrm{syn}}(H, w)$ denote the explicit syndetic return modulus from Lemma 14.6. Set

$$L_0 := L_{\mathrm{syn}}(H, w) + |w|. \tag{5}$$

Lemma 8.2 (No long tight runs in H can avoid w)**.** *Assume **(RC1)** holds for (H, w) and let L_0 be as in (5). If $\gamma = (e_0, \dots, e_{L-1})$ is a finite path that stays inside H and uses only tight edges $e_t \in E^{(0)}$, then γ contains an occurrence of w provided $L \geq L_0$. Equivalently, any tight path segment inside H with no occurrence of w has length at most $L_0 - 1$.*

Proof. Because H is strongly connected and contains a directed cycle, any finite tight path segment inside H can be extended forward to an infinite tight path in H. Apply Lemma 14.6 to that infinite tight extension: it contains occurrences of w with gaps between *starting indices* bounded by $L_{\mathrm{syn}}(H, w)$. In particular, there exists an occurrence of w whose start index is at most $L_{\mathrm{syn}}(H, w)$. Since an occurrence of w occupies $|w|$ consecutive edges, the entire occurrence lies within the first $L_{\mathrm{syn}}(H, w) + |w| = L_0$ edges. Thus any tight segment of length $L \geq L_0$ must contain an occurrence of w. □

8.3 Linear slack forcing on word-avoiding windows

We now turn Lemma 8.2 into a quantitative lower bound on slack.

Proposition 8.3 (Avoiding w for length L forces linear slack)**.** *Let H be a tight SCC of $G^{(0)}$ and let w be a tight word supported in H such that **(RC1)** holds for (H, w). Let L_0 be as in (5). Consider any finite path segment $\gamma = (e_0, \dots, e_{L-1})$ that stays inside H and contains* no *occurrence of w. Then*

$$\mathrm{SlackCount}(\gamma) \;\geq\; \max\left\{0,\; \left\lceil \frac{L - L_0}{L_0 + 1} \right\rceil \right\}, \tag{6}$$

and consequently, if $\alpha < \infty$,

$$\mathrm{SlackMass}(\gamma) \;\geq\; \alpha \cdot \max\left\{0,\; \left\lceil \frac{L - L_0}{L_0 + 1} \right\rceil \right\}. \tag{7}$$

In particular, $\mathrm{SlackMass}(\gamma)$ *grows at least linearly in L once $L > L_0$.*

Proof. If γ contains no tight edges, then $\mathrm{SlackCount}(\gamma) = L$ and (6) holds. Assume γ contains at least one tight edge. Decompose γ into maximal consecutive blocks of tight edges (tight runs) separated by slack edges. Let k be the number of tight runs and let $s = \mathrm{SlackCount}(\gamma)$ be the number of slack edges. Then necessarily $k \leq s + 1$.

By Lemma 8.2, every tight run inside H that avoids w has length at most $L_0 - 1$. Hence the total number of tight edges in γ is at most $k(L_0 - 1)$. Therefore

$$L = (\text{number of tight edges}) + s \;\leq\; k(L_0 - 1) + s \;\leq\; (s + 1)(L_0 - 1) + s = s(L_0 + 1) + (L_0 - 1).$$

Rearranging gives $s \geq (L - L_0)/(L_0 + 1)$. Since s is an integer and $s \geq 0$, we obtain (6). Finally, (7) follows from $\mathrm{SlackMass}(\gamma) \geq \alpha\, \mathrm{SlackCount}(\gamma)$. □

Remark 8.4 (What this does and does not say)**.** *Proposition 8.3 is a* local forcing *statement: it applies to windows that stay inside a fixed tight SCC H where **(RC1)** holds for a chosen word w. If **(RC1)** fails, Lemma 14.3 provides a tight periodic witness that avoids w with zero slack, showing that no positive lower bound of the form (7) can hold in that case. Excursions that leave H and later return are handled separately in Section 9.*

9 Return Cost for Excursions Out of a Tight SCC

This section isolates a second forcing mechanism that complements Section 8: even if a trajectory attempts to avoid a certified Doeblin word by *leaving* the relevant tight SCC, any later *return* to that SCC forces reduced-cost slack. This turns repeated excursions (a natural avoidance strategy) into a measurable slack burden.

9.1 A structural fact: tight SCCs do not admit tight returns from outside

Fix the calibrated potential ϕ and reduced costs c_ϕ. Let $G^{(0)}$ be the tight graph. Let H be an SCC of $G^{(0)}$.

Definition 9.1 (Excursion and return)**.** *Let $\pi = (e_0, e_1, \dots)$ be a one-sided path in the full graph G with vertex sequence $(v_t)_{t \geq 0}$, where $e_t : v_t \to v_{t+1}$. A time $t \geq 1$ is called a* return time *to H if $v_t \in H$ and $v_{t-1} \notin H$. An* excursion *is a maximal time interval during which $v_t \notin H$ following a departure from H and ending at the next return time (if it exists).*

Lemma 9.2 (Returning to a tight SCC forces slack). *Let $\gamma = (e_0, \dots, e_{L-1})$ be a finite path segment in G with vertex sequence $(v_0, \dots, v_L)$. Assume $v_0 \in H$ and $v_L \in H$, and assume that $v_t \notin H$ for at least one intermediate time t with $0 < t < L$. Then γ contains at least one non-tight edge, i.e.* $\mathrm{SlackCount}(\gamma) \geq 1$. *In particular, if $\alpha < \infty$ (as in (4)), then* $\mathrm{SlackMass}(\gamma) \geq \alpha$.

Proof. Suppose for contradiction that every edge of γ is tight, so γ is a path in the tight graph $G^{(0)}$. Then in $G^{(0)}$ there exists a tight path from H to a vertex outside H (since some $v_t \notin H$ occurs) and a tight path back from outside to H (since $v_L \in H$). This implies that in the SCC condensation DAG of $G^{(0)}$ there is a directed path from the SCC H to a different SCC and a directed path back to H, creating a directed cycle in the condensation. But the SCC condensation is a DAG, so such a cycle is impossible. Therefore γ must contain a non-tight edge, proving $\mathrm{SlackCount}(\gamma) \geq 1$. The bound on SlackMass follows from $\mathrm{SlackMass}(\gamma) \geq \alpha\, \mathrm{SlackCount}(\gamma)$. ☐

9.2 Quantifying slack per return

Proposition 9.3 (Slack lower bound per number of returns). *Let $\pi = (e_0, e_1, \dots)$ be any one-sided path in G and let $R_n(H)$ be the number of return times to H among $\{1, \dots, n\}$ (Definition 9.1). Then for every $n \geq 1$,*

$$R_n(H) \;\leq\; \sum_{t=0}^{n-1} \mathbf{1}_{\{e_t \notin E^{(0)}\}}, \tag{8}$$

and consequently, if $\alpha < \infty$,

$$\sum_{t=0}^{n-1} c_\phi(e_t) \;\geq\; \alpha \cdot R_n(H). \tag{9}$$

Proof. Each return time to H completes an excursion that starts in H, leaves H, and ends in H. Apply Lemma 9.2 to the finite path segment spanning that excursion. Different excursions are edge-disjoint in time, so each return time accounts for at least one distinct non-tight edge. This yields (8), and (9) follows from (4). ☐

9.3 Connection to word-avoidance strategies

Let H be a tight SCC and let w be a tight word supported in H. If **(RC1)** holds for (H, w), then Section 8 shows that long word-avoidance windows *inside* H force linear slack. Proposition 9.3 shows that word-avoidance by repeated *excursions* out of H also forces slack.

Remark 9.4 (Excursions as avoidance). *Assume **(RC1)** holds for (H, w) and let L_0 be as in (5). If a path visits H frequently but attempts to avoid w, then it must do at least one of the following on most long windows:*

- *introduce slack edges while staying in H, which is quantitatively penalized by Proposition 8.3;*
- *leave H before a tight run of length L_0 can occur and later return, which is penalized by Proposition 9.3.*

This is the mechanism used in the forcing dichotomy that follows: either w recurs with bounded gaps on the relevant tight core, or slack density is bounded below by a positive constant (contradicting asymptotic tightness for mean-minimizers).

10 Forcing Dichotomy: Doeblin Frequency versus Slack Density

This section upgrades the qualitative "tight-density" information available for general minimizers into a quantitative forcing statement. It combines:

- the **RC1** avoidance certificate for a word w inside a fixed tight aperiodic SCC H (Definition 14.1), and
- the **return-cost** principle for excursions out of H (Proposition 9.3).

The output is a dichotomy that is strong enough to feed the Doeblin⇒Hilbert-contraction chain: either the Doeblin word occurs with *uniform positive lower density* along the trajectory, or the reduced-cost slack has *positive lower density* (hence the trajectory cannot be minimizing).

10.1 Setup

Fix the calibrated potential ϕ and reduced costs $c_\phi(e) \geq 0$ as in (2). Let $E^{(0)} = \{e \in E : c_\phi(e) = 0\}$ be the tight edge set and let $G^{(0)} = (V^{(0)}, E^{(0)})$ be the tight graph.

Fix a tight SCC H of $G^{(0)}$ and a finite edge-word w supported in H such that **(RC1)** holds for (H, w). Let $L_{\mathrm{syn}}(H, w)$ be the explicit syndetic modulus from Lemma 14.6 and set

$$L_0 := L_{\mathrm{syn}}(H, w) + |w| \tag{10}$$

as in (5). Let

$$\alpha := \min\{c_\phi(e) : e \in E \setminus E^{(0)}\} \in (0, \infty] \tag{11}$$

as in (4).

For a one-sided path $\pi = (e_0, e_1, \dots)$ in G define the length-n slack mass

$$\mathrm{SlackMass}_n(\pi) := \sum_{t=0}^{n-1} c_\phi(e_t). \tag{12}$$

Define the tight time count in H by

$$T_H(n; \pi) := \sum_{t=0}^{n-1} \mathbf{1}_{\{e_t \in E^{(0)} \text{ and } e_t \text{ is supported in } H\}}, \tag{13}$$

and let $R_n(H)$ be the number of return times to H among $\{1, \dots, n\}$ as in Definition 9.1.

Finally, let $N_w(n; \pi)$ denote the number of starting indices $0 \leq i \leq n-1$ at which the word w occurs along π (i.e. $e_i e_{i+1} \cdots e_{i+|w|-1} = w$).

10.2 A counting lemma on maximal tight runs

Definition 10.1 (Maximal tight-in-H runs)**.** *For fixed n, consider the set of indices $t \in \{0, \dots, n-1\}$ for which e_t is tight and supported in H. The maximal contiguous intervals of such indices determine disjoint finite subpaths*

$$\gamma^{(1)}, \dots, \gamma^{(m)}$$

which we call the maximal tight-in-H runs *of π up to time n. Write $\ell_j := |\gamma^{(j)}|$ for the length (number of edges) of the j-th run. Then $\sum_{j=1}^m \ell_j = T_H(n; \pi)$ and $m - 1 \leq R_n(H)$.*

Lemma 10.2 (Each long tight run forces many occurrences of w)**.** *Assume **(RC1)** holds for (H, w) and let L_0 be as in (10). Let γ be a finite tight-in-H run of length ℓ. Then γ contains at least $\lfloor \ell/L_0 \rfloor$ occurrences of w.*

Proof. Partition γ into consecutive disjoint blocks of length L_0, discarding at most a final remainder block of length $< L_0$. Each full block is a tight path in H of length L_0. By Lemma 8.2, no such block can avoid w, so each full block contains an occurrence of w supported entirely inside that block. Disjoint blocks cannot share an occurrence, hence the number of occurrences is at least the number of full blocks, which is $\lfloor \ell/L_0 \rfloor$. □

Lemma 10.3 (Occurrences versus returns)**.** *For every $n \geq 1$ and every path π,*

$$N_w(n;\pi) \;\geq\; \frac{T_H(n;\pi)}{L_0} \;-\; R_n(H) \;-\; 1. \tag{14}$$

Proof. Let $\gamma^{(1)}, \dots, \gamma^{(m)}$ be the maximal tight-in-H runs up to time n (Definition 10.1) with lengths $\ell_1, \dots, \ell_m$. By Lemma 10.2,

$$N_w(n;\pi) \;\geq\; \sum_{j=1}^{m} \left\lfloor \frac{\ell_j}{L_0} \right\rfloor \;\geq\; \sum_{j=1}^{m} \left(\frac{\ell_j}{L_0} - 1 \right) \;=\; \frac{T_H(n;\pi)}{L_0} - m.$$

Since $m \leq R_n(H) + 1$, we obtain (14). □

10.3 The forcing dichotomy

Proposition 10.4 (Doeblin frequency vs slack density)**.** *Assume **(RC1)** holds for (H, w) and let L_0 and α be as in (10)–(11). Then for every $n \geq 1$ and every one-sided path π in G,*

$$N_w(n;\pi) \;\geq\; \frac{T_H(n;\pi)}{L_0} \;-\; \frac{\mathrm{SlackMass}_n(\pi)}{\alpha} \;-\; 1, \tag{15}$$

with the convention that $\mathrm{SlackMass}_n(\pi)/\alpha = 0$ if $\alpha = \infty$. Equivalently,

$$\mathrm{SlackMass}_n(\pi) \;\geq\; \alpha \cdot \left(\frac{T_H(n;\pi)}{L_0} - N_w(n;\pi) - 1 \right)_+. \tag{16}$$

Proof. Combine Lemma 10.3 with the slack-per-return bound (9) from Proposition 9.3, which yields $R_n(H) \leq \mathrm{SlackMass}_n(\pi)/\alpha$ (for $\alpha < \infty$). If $\alpha = \infty$, then $E \setminus E^{(0)} = \emptyset$ and $\mathrm{SlackMass}_n(\pi) = 0$ identically, so (15) reduces to (14). □

Corollary 10.5 (Forced positive density of the Doeblin word)**.** *Assume **(RC1)** holds for (H, w) and let L_0 be as in (10). Let π be any path such that*

$$\liminf_{n\to\infty} \frac{T_H(n;\pi)}{n} \;\geq\; \theta > 0 \qquad \textit{and} \qquad \lim_{n\to\infty} \frac{\mathrm{SlackMass}_n(\pi)}{n} \;=\; 0.$$

Then

$$\liminf_{n\to\infty} \frac{N_w(n;\pi)}{n} \;\geq\; \frac{\theta}{L_0}.$$

In particular, w occurs along π with a uniform positive lower density depending only on the forcing constants.

Proof. Divide (15) by n and take $\liminf_{n\to\infty}$, using the hypotheses. □

Remark 10.6 (Interpretation)**.** *Corollary 10.5 is the promised forcing upgrade: in the regime where a minimizer has nontrivial tight mass in a fixed aperiodic tight SCC H and reduced-cost slack density* 0*,* ***(RC1)*** *forces a quantitative supply of Doeblin blocks along the trajectory. This is sufficient for explicit exponential projective contraction (see the rate pack in Section 18), without needing global bounded-gap recurrence.*

11 Descriptor Faces as Finite Certificates (Eliminating Interface Assumptions)

Two interface notions recur throughout the gap-free theory on a tight SCC H: *active-face invariance* and *support-connectivity on the active face.* In this section we turn both into purely *finite, checkable certificates* extracted directly from the reduced descriptor graph and template sparsity.

11.1 Canonical active face per descriptor state

In the descriptor reduction, each vertex $v \in V$ corresponds to a finite descriptor encoding which coordinates are *potentially active* and which are structurally zero under admissible propagation. Accordingly, each descriptor vertex carries a canonical support set

$$S(v) \subseteq \{1, \dots, d\}.$$

Definition 11.1 (Face map and canonical active set)**.** *A* face map *for the reduced graph is a function $S : V \to 2^{\{1,\dots,d\}}$ such that for each edge $e : u \to v$ labeled by a template (or fixed-length block) T_e we have:*

(i) ***(Forward invariance)*** $T_e(\mathbb{R}^{S(u)}_{>0}) \subseteq \mathbb{R}^{S(v)}_{>0}$*;*

(ii) ***(No spurious activation)*** *If $x \in \mathbb{R}^{S(u)}_{>0}$, then $(T_e x)_i = 0$ for all $i \notin S(v)$.*

Remark 11.2. *Definition 11.1 is checkable from template sparsity: it amounts to verifying that the support pattern of each T_e maps the indicator of $S(u)$ into $S(v)$.*

11.2 Face stability on SCCs as a finite fixed-point

Lemma 11.3 (SCC face stability by intersection closure)**.** *Let H be an SCC of the reduced graph and let $S : V \to 2^{\{1,\dots,d\}}$ be a face map. Define the SCC face by*

$$S(H) := \bigcap_{v \in H} S(v).$$

Then for every edge $e : u \to v$ inside H,

$$T_e(\mathbb{R}^{S(H)}_{>0}) \subseteq \mathbb{R}^{S(H)}_{>0}.$$

Proof. Since $S(H) \subseteq S(u)$ for each $u \in H$, we have $\mathbb{R}^{S(H)}_{>0} \subseteq \mathbb{R}^{S(u)}_{>0}$. By forward invariance, $T_e(\mathbb{R}^{S(u)}_{>0}) \subseteq \mathbb{R}^{S(v)}_{>0}$. Because $S(H) \subseteq S(v)$, restricting to coordinates in $S(H)$ yields invariance on $S(H)$. □

11.3 Support-connectivity on the SCC face as a finite graph property

Fix an SCC H and its canonical face $S(H)$. For each generator edge e in H labeled by T_e, define the coordinate support digraph on $S(H)$: we draw an arrow $j \to i$ if there exists an edge e in H with $(T_e)_{ij} > 0$. Thus $j \to i$ records that coordinate j can contribute to coordinate i in one admissible step inside H.

Definition 11.4 (Coordinate support digraph on a face)**.** *Let $\Gamma(H)$ be the directed graph with vertex set $S(H)$ and an arrow $j \to i$ if there exists an edge e in H such that $(T_e)_{ij} > 0$.*

Lemma 11.5 (Support-connectivity certificate)**.** *If $\Gamma(H)$ is strongly connected, then for every $i, j \in S(H)$ there exists an admissible word w in H such that the block product satisfies $(B(w))_{ij} > 0$.*

Proof. Strong connectivity gives a coordinate path $j \to \cdots \to i$ realized by a sequence of generator matrices from edges of H. Concatenate the corresponding edges to obtain a word w whose product contains a positive contribution from j into i. □

Corollary 11.6 (Doeblin word from aperiodic SCC via Boolean-pattern product automaton)**.** *Let H be an aperiodic SCC of the tight graph $G^{(0)}$ and assume a consistent face map $S(\cdot)$ is available, with active face $S(H) \neq \varnothing$. Then the following are equivalent:*

(i) There exists a finite word w supported in H such that the block product $B(w)$ is strictly positive on $S(H) \times S(H)$ (equivalently, w is Doeblin on the face $S(H)$).

(ii) The Boolean-pattern product-automaton test (Algorithm 6) reaches a return state $(v_0, \mathbf{1})$ for some base vertex $v_0 \in H$, where $\mathbf{1}$ denotes the all-ones pattern on $S(H) \times S(H)$; in this case the algorithm returns an explicit witness word w.

In case (i), strict positivity of $B(w)_{S(H)\times S(H)}$ implies that for every $i, j \in S(H)$ there exists a directed coordinate path $j \to \cdots \to i$ in $\Gamma(H)$, hence $\Gamma(H)$ is strongly connected. Thus failure of strong connectivity produces the OG3 cut witness of Lemma 11.10, and strong connectivity of $\Gamma(H)$ is a fast necessary check. When $\Gamma(H)$ is strongly connected, the existence of a Doeblin word is decided by the Boolean-pattern product-automaton test (Algorithm 6); the graph condition alone does not certify strict positivity on $S(H) \times S(H)$.

Remark 11.7 (Strong connectivity is necessary but not sufficient)**.** *Strong connectivity of $\Gamma(H)$ is a fast necessary test for the existence of a Doeblin word. However, $\Gamma(H)$ records only one-step support reachability and does not encode higher-order pattern constraints that can prevent simultaneous positivity of all pairs. A clean counter-shape is the permutation case: if all allowed face-restricted patterns are permutation matrices generating a transitive action on $S(H)$, then $\Gamma(H)$ is strongly connected while every admissible product remains a permutation pattern, so no word yields strict positivity on $S(H) \times S(H)$. For this reason, the pipeline uses the Boolean-pattern product-automaton test (Algorithm 6) as the definitive finite decision procedure; when it fails, it returns an explicit finite non-reachability certificate in that automaton.*

11.4 Algorithmic extraction

11.5 Witness extraction when certificates fail

The obstruction predicates OG2/OG3 are designed to be not merely *named* failures but *witness-producing* ones. Here we record explicit finite witnesses that can be returned whenever the face-map or coordinate-connectivity checks fail.

Definition 11.8 (Support propagation operator)**.** *For an edge $e : u \to v$ labeled by a nonnegative template/block T_e and a set $S \subseteq \{1, \dots, d\}$, define the propagated support*

$$\mathrm{Prop}_e(S) := \{\, i : \ \exists j \in S \text{ with } (T_e)_{ij} > 0 \,\}.$$

For a word $w = e_1 \cdots e_L$ we set $\mathrm{Prop}_w := \mathrm{Prop}_{e_L} \circ \cdots \circ \mathrm{Prop}_{e_1}$.

Lemma 11.9 (OG2 witness: finite inconsistency cycle)**.** *Let H be an SCC and suppose the proposed face sets $S(v)$ fail to form a face map on H. Then there exist vertices $u, v \in H$, an edge $e : u \to v$ in H, and indices $j \in S(u)$ and $i \notin S(v)$ such that $(T_e)_{ij} > 0$. Moreover, choosing any directed path p in H from v back to u and setting $w := ep$, we obtain a directed cycle word w in H together with a marked step (the first step e) at which a coordinate outside the designated face is activated. This pair $(w, (i, j))$ is a finite witness of OG2 on H.*

Proof. If $S(\cdot)$ is not a face map on H, then some edge $e : u \to v$ in H violates Definition 11.1. In particular, the "no spurious activation" condition (ii) fails, which means exactly that $\mathrm{Prop}_e(S(u)) \nsubseteq S(v)$. Equivalently, there exist $j \in S(u)$ and $i \notin S(v)$ with $(T_e)_{ij} > 0$. Because H is strongly connected, there exists a directed path p from v to u, so $w := ep$ is a directed cycle word in H. The marked first step e certifies the activation of an outside coordinate along a cycle in H. □

Lemma 11.10 (OG3 witness: cut decomposition of $\Gamma(H)$)**.** *Let H be a tight SCC for which a face map exists and let $S(H)$ be the SCC face. If the coordinate digraph $\Gamma(H)$ is not strongly connected, then there exist nonempty disjoint sets $A, B \subseteq S(H)$ with $A \cup B = S(H)$ such that*

$$(T_e)_{ij} = 0 \qquad \text{for every edge } e \text{ in } H, \text{ every } i \in A, \text{ and every } j \in B.$$

Equivalently, $\Gamma(H)$ has no directed edges from B to A. In particular, for every admissible word w in H and every $i \in A$, $j \in B$ we have $(B(w))_{ij} = 0$. Thus $(H, S(H), A, B)$ is a finite witness of OG3.

Proof. Compute the SCC decomposition of $\Gamma(H)$ and consider its condensation DAG. Since $\Gamma(H)$ is not strongly connected, this DAG has at least two nodes and hence has a sink SCC A. Let $B := S(H) \setminus A$. By the definition of a sink SCC, there is no directed edge in $\Gamma(H)$ from any vertex in B to any vertex in A. By Definition 11.4, an edge $j \to i$ in $\Gamma(H)$ exists exactly when some generator edge e in H has $(T_e)_{ij} > 0$. Therefore for all $i \in A$ and $j \in B$, and for all edges e in H, we have $(T_e)_{ij} = 0$. The final structural-zero claim for products follows by closure under multiplication of nonnegative matrices: if every generator has zeros on $A \times B$, then every product does as well. □

Algorithm 4 Compute SCC face and coordinate support graph

Require: SCC H with face map $S(\cdot)$ and generator labels T_e
Ensure: $S(H)$ and coordinate digraph $\Gamma(H)$

```
S(H) ← ⋂_{v∈H} S(v)
Initialize Γ(H) with vertex set S(H)
for all edges e inside H do
    for all i, j ∈ S(H) do
        if (T_e)_ij > 0 then add arrow j → i
        end if
    end for
end for
```

Algorithm 5 Check OG2–OG3 and produce Doeblin/recurrence witnesses

Require: Tight graph $G^{(0)}$, SCC decomposition, period and coordinate digraphs $\Gamma(H)$
Ensure: Either a finite obstruction witness (OG1/OG2/OG3), or a Doeblin witness object (a word w or a canonical family $\mathcal{B}(H)$) together with either a syndetic modulus L_{syn} (RC1/MB–RC1) or an explicit avoidance-cycle witness

Compute SCCs of $G^{(0)}$ and their periods $\text{per}(H)$
for each aperiodic SCC H **do**
    Attempt to compute a consistent face map $S(\cdot)$ on H
    if inconsistent **then**
        return OG2 witness (inconsistency cycle)
    end if
    Build $\Gamma(H)$ on $S(H)$
    if $\Gamma(H)$ is not strongly connected **then**
        return OG3 witness of type (cut) via Lemma 11.10
    end if
    Run the product-automaton Doeblin-word test (Algorithm 6)
    if Algorithm 6 returns a word w **then**
        Compute the canonical family $\mathcal{B}(H)$ (Section 13)
        if canonical multi-block mode is enabled **then**
            Build the multi-block avoidance graph $\mathsf{A}(H, \mathcal{B}(H))$
            if $\mathsf{A}(H, \mathcal{B}(H))$ contains a directed cycle **then**
                return OG4^{MB} witness (avoidance-cycle witness in $\mathsf{A}(H, \mathcal{B}(H))$)
            else
                Compute $L_{\text{syn}} = 1 + \ell_{\max}$ where $\ell_{\max}$ is the longest path length in $\mathsf{A}(H, \mathcal{B}(H))$
                return $(\mathcal{B}(H), L_{\text{syn}})$ (MB–RC1; Definition 13.5)
            end if
        end if
        Build the avoidance graph $\mathcal{A}_{\neg w}(H)$ (Definition 14.2)
        if $\mathcal{A}_{\neg w}(H)$ contains a directed cycle **then**
            return Doeblin witness w and an avoidance-cycle witness (Lemma 14.3)
        else
            Compute $L_{\text{syn}} = 1 + \ell_{\max}$ where $\ell_{\max}$ is the longest path length in $\mathcal{A}_{\neg w}(H)$
            return (w, L_{syn}) (RC1; Lemma 14.6)
        end if
    else
        return OG3 witness of type (product-automaton): a finite non-reachability certificate in the Boolean-pattern product automaton (no return state $(v_0, \mathbf{1})$ is reachable)
    end if
end for
return OG1 witness (no aperiodic SCC exists in $G^{(0)}$)

12 Doeblin-word discovery by a finite product automaton

This section records the finite-state procedure used throughout the certificate pipeline to decide whether a given aperiodic tight SCC admits a Doeblin word on its active face. The input is purely combinatorial support information on the face, and the output is either a concrete witness word or a finite non-reachability certificate.

12.1 Boolean patterns and Boolean product

Fix a tight SCC H and a face $S = S(H) \subseteq \{1, \ldots, d\}$. For each tight edge $e : u \to v$ in H, let $B_e \in \mathbb{R}_+^{d\times d}$ denote the corresponding nonnegative block (or fixed-length template). Define the face-restricted support pattern

$$P_e \in \{0,1\}^{S\times S}, \qquad (P_e)_{ij} = 1 \iff (B_e)_{ij} > 0 \text{ for } i, j \in S.$$

For Boolean patterns $P, Q \in \{0,1\}^{S\times S}$, define their Boolean product $P \odot Q$ by

$$(P \odot Q)_{ij} = \bigvee_{k\in S} \big(P_{ik} \wedge Q_{kj}\big),$$

so that $P \odot Q$ records the support pattern of the matrix product of the underlying nonnegative blocks restricted to S.

12.2 The product automaton and BFS search

Let $\mathcal{P} := \{0,1\}^{S\times S}$ be the finite set of Boolean patterns. The product automaton has state space $H \times \mathcal{P}$. A directed transition

$$(u, P) \xrightarrow{e:u\to v} (v, P_e \odot P)$$

records the effect of appending the edge e to the front of the word whose current pattern is P.

Algorithm 6 Doeblin-word discovery by finite product automaton

---

Require: Tight SCC H, active face $S(H)$, Boolean support patterns P_e on $S(H)$ for each tight edge e in H

Ensure: Either a word w whose face-restricted product is strictly positive, or a finite non-reachability certificate

Choose the canonical base vertex $v_0 := v_{\min}(H)$ (Definition 13.1)
Let I be the identity pattern on $S(H)$ and let $\mathbf{1}$ denote the all-ones pattern on $S(H) \times S(H)$
Initialize BFS on states (v, P) with the start state (v_0, I)
while queue not empty **do**
  Pop (u, P)
  for all tight edges $e : u \to v$ in H, iterated in canonical outgoing-edge order (Definition 13.1) **do**
    $P' \leftarrow P_e \odot P$
    if (v, P') not yet visited **then**
      Mark visited; record predecessor; enqueue (v, P')
    end if
    if $v = v_0$ **and** $P' = \mathbf{1}$ **then**
      Reconstruct the witnessed word w from predecessors and **return** w
    end if
  end for
end while
return finite non-reachability certificate: $\mathbf{1}$ is not reachable at $(v_0, \cdot)$

Proposition 12.1 (Shortest Doeblin witness and a state-size length bound)**.** *If Algorithm 6 returns a witness word w, then w has minimal length among all edge words in H based at v_0 whose face-restricted Boolean product equals the all-ones pattern. Moreover, one has the explicit bound*

$$|w| \leq |H| \cdot 2^{|S(H)|^2} - 1,$$

since the BFS explores at most $|H| \cdot 2^{|S(H)|^2}$ states.

Proof. Breadth-first search discovers states in nondecreasing path length. The first time the target condition $v = v_0$ and $P = \mathbf{1}$ is met, the predecessor pointers reconstruct a shortest path in the product automaton, hence a shortest witness word. The length bound follows because a shortest path in a finite graph does not repeat a state, and the automaton has $|H| \cdot 2^{|S(H)|^2}$ states. □

Remark 12.2 (Complexity)**.** *The state space has size $|H| \cdot 2^{|S(H)|^2}$. This exponential dependence on the face size is unavoidable at the level of brute-force semigroup reachability, but in the intended regime the active faces are descriptor-controlled and small.*

Remark 12.3 (Language-aware reachability versus conservative support filters)**.** *In some implementations it is convenient to precompute auxiliary reachability graphs from template support information that are (intentionally) conservative. Such over-approximations may make the presentation look looser than necessary, but they do not affect correctness: the decisive Doeblin step is the product-automaton decision above, which is built from the actual tight-language edges in H and the face-restricted patterns P_e. Any word returned by Algorithm 6 is a valid witness on the declared face, and any non-reachability conclusion is only asserted for the declared automaton.*

13 Canonical Doeblin block families and multi-block shell rigidity

The RC1/OG4 layer in Section 14 is stated for a *fixed* certified Doeblin word w. For maximal universality inside a fixed operator class, it is advantageous to make the choice of contraction block(s) canonical, then certify recurrence against a *finite family* of contraction blocks rather than a single analyst-chosen word.

The purpose of this section is therefore twofold:

- define a deterministic procedure that selects a canonical finite Doeblin family $\mathcal{B}(H)$ on the active face of a tight aperiodic SCC H;
- define a multi-block recurrence predicate **(MB–RC1)** whose failure is recorded as a refined OG4 witness.

13.1 Canonical selection rule for a Doeblin family

Fix a tight aperiodic SCC H and its active face $S = S(H)$. Algorithm 6 returns a shortest Doeblin witness based at a base vertex $v_0 \in H$. To remove analyst choice, we fix the following canonical conventions.

Definition 13.1 (Canonical basepoint and edge ordering)**.** *Assume the certificate data provides a canonical total order on vertices $V(H)$ and edges $E(H)$ (e.g. by stable integer IDs in the exported model). Define the canonical basepoint to be*

$$v_{\min}(H) := \min V(H),$$

and order outgoing edges at each vertex by increasing edge ID. All BFS/DFS routines in the certificate pipeline use these canonical orders.

Definition 13.2 (Canonical Doeblin family). *Let $w_\star(H)$ be the shortest Doeblin witness returned by Algorithm 6 when run with basepoint $v_{\min}(H)$ and canonical outgoing-edge iteration. Let $K := |w_\star(H)|$. Define the* canonical Doeblin family

$$\mathcal{B}(H) := \{\, w : \ w \text{ is a Doeblin witness in } H \text{ based at } v_{\min}(H) \text{ with } |w| = K \,\},$$

listed in lexicographic order by edge ID sequence. (Equivalently, $\mathcal{B}(H)$ is the set of all shortest Doeblin witnesses based at the canonical basepoint.)

Remark 13.3 (Why a family). *A single shortest witness is already canonical under Definition 13.1. However, using the full finite family $\mathcal{B}(H)$ strengthens the shell-rigidity check: MB–RC1 certifies that* some *certified contraction block from the canonical family must occur with bounded gaps, which is strictly stronger than recurrence of one chosen word and reduces sensitivity to accidental symmetries.*

13.2 Aho–Corasick dictionary automaton and avoidance graph

Let $\mathcal{B} = \mathcal{B}(H)$ and let $\Sigma := E(H)$ be the alphabet of tight edges. Consider the standard Aho–Corasick (dictionary) automaton built on the pattern set $\mathcal{B}$. It has a finite state set $Q_{\mathrm{AC}}(\mathcal{B})$ encoding the longest suffix of the read word that is a prefix of some pattern in $\mathcal{B}$, together with failure links. A state is *accepting* if some $w \in \mathcal{B}$ has just occurred.

Definition 13.4 (Multi-block avoidance graph). *Let $\mathsf{A}(H, \mathcal{B})$ be the directed graph on vertex set*

$$V(H) \times Q_{\mathrm{AC}}(\mathcal{B})_{\mathrm{nonacc}},$$

where $Q_{\mathrm{AC}}(\mathcal{B})_{\mathrm{nonacc}}$ denotes the non-accepting states. There is a directed edge

$$(x, q) \to (y, q')$$

iff there exists a tight edge $e : x \to y$ in H such that the Aho–Corasick update sends $q \xrightarrow{e} q'$ and q' is non-accepting. Thus $\mathsf{A}(H, \mathcal{B})$ encodes the evolution along paths in H under the constraint that no pattern in $\mathcal{B}$ has occurred.

13.3 Multi-block shell rigidity (MB–RC1)

Definition 13.5 (Multi-block recurrence input (MB–RC1)). *Let H be a tight aperiodic SCC and let $\mathcal{B}(H)$ be the canonical Doeblin family. We say that **(MB–RC1)** holds for H if the multi-block avoidance graph $\mathsf{A}(H, \mathcal{B}(H))$ is acyclic. In that case there exists a finite modulus $L_{\mathrm{syn}}(H, \mathcal{B}(H))$ such that every length-$L_{\mathrm{syn}}(H, \mathcal{B}(H))$ tight word in H contains* at least one *block from $\mathcal{B}(H)$.*

Lemma 13.6 (Acyclicity forces bounded-gap block hits). *If $\mathsf{A}(H, \mathcal{B})$ is acyclic, then there is a finite integer $L_{\mathrm{syn}}(H, \mathcal{B})$ such that every tight edge word in H of length $L_{\mathrm{syn}}(H, \mathcal{B})$ contains an occurrence of some $w \in \mathcal{B}$. Equivalently, every infinite tight trajectory in H hits the set of words $\mathcal{B}$ with gaps at most $L_{\mathrm{syn}}(H, \mathcal{B})$.*

Proof. Along any finite tight word in H that avoids *all* patterns in $\mathcal{B}$, the lifted trajectory stays in the non-accepting states, so it corresponds to a directed path in $\mathsf{A}(H, \mathcal{B})$. If the avoidance graph is acyclic, its longest directed path length is finite; call it $\ell_{\max}$. Then any word of length $> \ell_{\max}$ would force repetition and hence a directed cycle, contradiction. Thus every block-avoiding word has length at most $\ell_{\max}$. Taking $L_{\mathrm{syn}} := \ell_{\max} + 1$ yields the claim. ☐

Remark 13.7 (RC1 as the single-block special case). *If $\mathcal{B} = \{w\}$ is a singleton, the Aho–Corasick automaton reduces to the usual prefix automaton and $\mathsf{A}(H, \mathcal{B})$ coincides with $\mathsf{A}(H, w)$ from Definition 14.2. Hence MB–RC1 reduces to RC1.*

Certificate fields. In the exported `RobustCertificate`, the Doeblin field records:

- the canonical basepoint $v_{\min}(H)$ and the canonical selection rule;
- the family $\mathcal{B}(H)$ (edge-id sequences) and the minimal Doeblin margin $\delta_{\min} := \min_{w \in \mathcal{B}(H)} \delta(w)$;
- the multi-block modulus $L_{\mathrm{syn}}(H, \mathcal{B}(H))$ whenever MB–RC1 holds.

These fields make the contraction mechanism deterministic and strengthen the shell-rigidity gate without changing the operator class.

14 Syndetic return to a certified Doeblin word via an avoidance graph

Fix a tight SCC H and a word $w = e_1 \cdots e_m$ on the edges of H. This section formalizes a finite, checkable recurrence condition for w. The outcome is dichotomous:

- either the avoidance dynamics contains a directed cycle, yielding an explicit witness that w can be avoided for arbitrarily long times,
- or the avoidance dynamics is acyclic, in which case w must occur with bounded gaps along every infinite trajectory in H.

The second case is the concrete recurrence input (RC1) used in the gap-free contraction statements.

Definition 14.1 (Finite recurrence input (RC1))**.** *Let H be a tight SCC and let w be an edge word in H. We say that (H, w) satisfies **(RC1)** if the avoidance graph $\mathsf{A}(H, w)$ (Definition 14.2) is acyclic. In this case Algorithm 7 returns a finite modulus $L_{\mathrm{syn}}(H, w)$ such that every edge word of length $L_{\mathrm{syn}}(H, w)$ in H contains an occurrence of w. For notational compatibility with later sections, we also write $\mathcal{A}_{\neg w}(H) := \mathsf{A}(H, w)$.*

14.1 Prefix automaton for a fixed word

Let $m := |w|$. For a finite edge word $a = e'_1 \cdots e'_k$ in H, define $\pi(a) \in \{0, 1, \ldots, m-1\}$ to be the length of the longest suffix of a which is a proper prefix of w. Equivalently, $\pi(a) = q$ means that the last q edges of a match $e_1 \cdots e_q$, but a does not yet contain a full occurrence of w ending at the last edge.

Given $q \in \{0, \ldots, m-1\}$ and an edge e of H, define the update $\mathrm{next}(q, e) \in \{0, \ldots, m\}$ to be the new matched-prefix length after appending e. If $\mathrm{next}(q, e) = m$, then w occurs ending at that step. Otherwise $\mathrm{next}(q, e) \in \{0, \ldots, m-1\}$. This is the standard prefix automaton update (KMP-type) for a fixed pattern.

14.2 Avoidance graph

Definition 14.2 (Avoidance graph)**.** *Let $\mathsf{A}(H, w)$ be the directed graph on vertex set*

$$\{0, 1, \ldots, m-1\} \times V(H).$$

There is a directed edge

$$(q, x) \to (q', y)$$

in $\mathsf{A}(H, w)$ if and only if there exists an edge $e : x \to y$ of H such that $\mathrm{next}(q, e) = q' \in \{0, \ldots, m-1\}$. In other words, $\mathsf{A}(H, w)$ encodes the evolution of the matched-prefix state along paths in H under the constraint that no step completes an occurrence of w.

Lemma 14.3 (Avoidance-cycle witness)**.** *If* $\mathsf{A}(H,w)$ *contains a directed cycle, then there exists an infinite trajectory in* H *which avoids* w *forever. Consequently, no bound on gaps between occurrences of* w *can hold uniformly over all infinite trajectories in* H*.*

Proof. A directed cycle in $\mathsf{A}(H,w)$ corresponds to a nonempty edge word in H which returns to the same matched-prefix state without ever completing w. Repeating this cycle yields an infinite path in H for which the prefix-state never reaches m, hence w never occurs. □

14.3 Sofic refinement: the w-avoidance shift

The avoidance graph also gives a finite-state symbolic model for the "bad" itineraries that avoid the certified word.

Definition 14.4 (w-avoidance shift)**.** *Let* $X_{\neg w}(H) \subset E(H)^{\mathbb{Z}}$ *be the set of bi-infinite edge sequences in* H *that avoid* w *at every time. Equivalently,* $X_{\neg w}(H)$ *is the two-sided edge shift of the avoidance graph* $\mathsf{A}(H,w)$*, viewed as a labeled graph by the underlying edges of* H*. In particular,* $X_{\neg w}(H)$ *is a sofic shift presented by a finite graph.*

Proposition 14.5 (RC1 versus nonemptiness of the avoidance shift)**.** *For fixed* (H,w)*, the following are equivalent:*

- $\mathsf{A}(H,w)$ *contains a directed cycle;*
- $X_{\neg w}(H)$ *is nonempty;*
- *OG4*(H,w) *holds (equivalently, RC1 fails).*

Consequently, RC1 holds if and only if $X_{\neg w}(H) = \varnothing$*, i.e. there are no bi-infinite tight itineraries in* H *that avoid* w*.*

Proof. A finite directed graph supports a bi-infinite path if and only if it contains a directed cycle. By Definition 14.4, bi-infinite w-avoiding itineraries in H correspond exactly to bi-infinite paths in $\mathsf{A}(H,w)$. The remaining equivalences are Definitions 14.1 and 16.1. □

Lemma 14.6 (Syndetic modulus from acyclicity)**.** *If* $\mathsf{A}(H,w)$ *is acyclic, then there exists a finite integer* $L_{\text{syn}}(H,w)$ *such that every length-*$L_{\text{syn}}(H,w)$ *edge word in* H *contains an occurrence of* w*. In particular, every infinite trajectory in* H *contains occurrences of* w *with gaps at most* $L_{\text{syn}}(H,w)$*.*

Proof. If $\mathsf{A}(H,w)$ is acyclic, it has finite longest directed path length. Along any edge word in H that avoids w, the corresponding lifted path in $\mathsf{A}(H,w)$ never hits an accepting step (next $= m$), so it remains a directed path in the avoidance graph. Therefore any w-avoiding word has length at most the longest path length of $\mathsf{A}(H,w)$. Taking $L_{\text{syn}}(H,w)$ to be one plus that maximal length forces every longer word to contain w. □

Corollary 14.7 (Explicit recurrence bound from automaton size)**.** *If* $\mathsf{A}(H,w)$ *is acyclic and has* N *vertices, then one may take*

$$L_{\text{syn}}(H,w) \le N.$$

In particular, since $N = m\,|V(H)|$*, the bounded-gap modulus can be chosen at most* $m\,|V(H)|$*.*

Proof. An acyclic directed graph on N vertices has no directed path longer than $N-1$ edges, so $\ell_{\max} \le N-1$ in Algorithm 7. Hence $L_{\text{syn}} = \ell_{\max} + 1 \le N$. □

Algorithm 7 Syndetic return check and modulus computation

Require: Tight SCC H, word w on edges of H of length m
Ensure: Either an avoidance-cycle witness, or a syndetic modulus $L_{\mathrm{syn}}(H, w)$
Build the avoidance graph $\mathsf{A}(H, w)$ (Definition 14.2)
if $\mathsf{A}(H, w)$ contains a directed cycle **then**
  Extract a directed cycle as a witness and **return** avoidance-cycle witness
else
  Compute the length $\ell_{\max}$ of a longest directed path in $\mathsf{A}(H, w)$ (DP on a topological ordering)
  return $L_{\mathrm{syn}}(H, w) \leftarrow \ell_{\max} + 1$
end if

15 Effective extremal language and a sofic Mather model

This section upgrades the "effective extremal itinerary" discussion to a fully witness-producing statement. The output is a finite symbolic object that can be validated independently, and it is the artifact associated with OP–MATHER–EFF.

15.1 From calibrated minimizers to a finite boundary carrier

Recall that canonical normalization and calibration assign reduced costs $c_\phi(e) \geq 0$ to edges and define the tight subgraph $G^{(0)}$ consisting of edges with $c_\phi(e) = 0$. The boundary carrier $\Sigma_{\mathcal{A}}$ (Definition 4.6) is the two-sided edge shift of a finite directed graph $G^{\mathcal{A}}$ extracted from the reduction.

Proposition 15.1 (Computable boundary carrier)**.** *Given a certified instance (Definition 1.2), the boundary carrier $\Sigma_{\mathcal{A}}$ is computable as a finite shift of finite type:*

- *the underlying graph $G^{\mathcal{A}}$ is obtained by selecting the tight edges and restricting to the active face data,*
- *the irreducible components of $\Sigma_{\mathcal{A}}$ are the SCC edge shifts of $G^{\mathcal{A}}$.*

Moreover, in the periodic regime of Proposition 7.3, each component is a single periodic orbit.

Proof. All objects in the construction are finite. The tight edge test is a finite comparison of reduced costs. Face restriction is a finite support check on the template blocks restricted to the declared active support sets. SCC decomposition is a finite graph computation. The final statement follows from Proposition 7.3. □

15.2 Sofic refinement from a Doeblin word and recurrence/avoidance

Fix a tight SCC $H \subseteq G^{\mathcal{A}}$. If w is a word supported in H, then the w-avoidance shift $X_{\neg w}(H)$ (Definition 14.4) is accepted by the finite avoidance automaton $\mathsf{A}(H, w)$. In particular, $X_{\neg w}(H) \neq \varnothing$ is witnessed by a directed cycle in $\mathsf{A}(H, w)$.

Definition 15.2 (Sofic extremal model associated to (H, w))**.** *Let H be a tight SCC and let w be a word supported in H. The* sofic extremal model *is the finite symbolic object consisting of:*

- *the edge shift X_H of H (a shift of finite type given by the adjacency of H),*
- *the avoidance automaton $\mathsf{A}(H, w)$,*

- *either an avoidance-cycle witness (a directed cycle in $\mathsf{A}(H, w)$), or a bounded-gap modulus $B(H, w)$ proving that every bi-infinite path in H contains an occurrence of w in every window of length $B(H, w)$.*

The bounded-gap modulus is the recurrence certificate RC1: when $\mathsf{A}(H, w)$ is acyclic, a finite longest-path computation yields $B(H, w)$.

Theorem 15.3 (Effective Mather/sofic description: periodic vs aperiodic regimes)**.** *For any certified instance, the pipeline returns a finite extremal itinerary model as follows.*

- ***Periodic regime.*** *If the minimizing carrier is minimal, then every tight SCC is a directed cycle (Proposition 7.3). In this case the extremal itinerary set is a finite union of periodic orbits, each specified by its cycle in $G^{\mathcal{A}}$.*
- ***Aperiodic regime with a candidate core.*** *If H is an aperiodic tight SCC and w is a certified Doeblin word on the associated face block, then the recurrence layer decides which of the following holds:*
  - ***RC1 holds:*** *$\mathsf{A}(H, w)$ is acyclic, a bounded-gap modulus $B(H, w)$ is computed, and every tight itinerary in H contains w syndetically.*
  - ***OG4 holds:*** *a directed cycle in $\mathsf{A}(H, w)$ is returned, yielding a periodic tight itinerary that avoids w forever.*

  In either subcase, the output is a concrete sofic extremal model in the sense of Definition 15.2, and it is verifiable directly from the returned finite data.

Consequently, OP–MATHER–EFF holds on the certified class: extremal itineraries admit an explicit finite-state model computed by the pipeline, and failure of the desired recurrence is witnessed by a finite avoidance cycle.

Proof. The periodic regime statement is Proposition 7.3. For the aperiodic regime, the Doeblin-word search (Section 12) produces either a word w with a positivity margin on the face block or a finite failure witness (OG3). Given (H, w), the avoidance automaton $\mathsf{A}(H, w)$ is finite and its cycle/acyclic dichotomy is decided by finite graph search. If acyclic, the maximal path length is finite and gives $B(H, w)$; if cyclic, the cycle itself yields a periodic w-avoiding itinerary. All returned objects are finite and checkable. □

15.3 Artifact: a self-contained symbolic model object

The paper includes a lightweight JSON schema for exporting the model in Definition 15.2. A verifier for the object is straightforward: validate the schema, then check that each listed transition belongs to the declared tight SCC, and that the avoidance-cycle witness (if present) is a genuine directed cycle.

16 Recurrence obstruction: shell-rigidity failure (MB–RC1 failure)

The obstruction taxonomy OG1–OG3 controls the existence of an aperiodic tight SCC, a consistent active-face map, and a strictly positive Doeblin block on that active face. Even when none of these obstructions occurs, a logically correct gap-free contraction statement must still address one further issue: a certified Doeblin word need not recur along *every* tight trajectory unless its recurrence is forced by finite-state structure.

We therefore isolate a finite recurrence predicate for a fixed SCC and a *canonical Doeblin family* (Section 13), and we record its failure as a fourth obstruction class OG4 with an explicit finite witness. This keeps recurrence out of the hypotheses while preserving a complete, decidable boundary taxonomy.

16.1 Definition and finite witness schema

Definition 16.1 (OG4 / recurrence obstruction: Doeblin-avoidance cycle)**.** *Let H be a tight SCC of $G^{(0)}$ and let w be a finite edge-word supported in H. We say that the obstruction* $\boldsymbol{OG4}(H,w)$ *holds if the avoidance graph $\mathcal{A}_{\neg w}(H)$ (Definition 14.2) contains a directed cycle. Equivalently,* ***(RC1)*** *fails for (H,w) (Definition 14.1).*

Definition 16.2 (OG4$^{\mathrm{MB}}$ / multi-block recurrence obstruction)**.** *Let H be a tight aperiodic SCC of $G^{(0)}$ and let $\mathcal{B}(H)$ be its canonical Doeblin family (Definition 13.2). We say that the obstruction* $\boldsymbol{OG4}^{\mathrm{MB}}(H)$ *holds if the multi-block avoidance graph $\mathsf{A}(H,\mathcal{B}(H))$ (Definition 13.4) contains a directed cycle. Equivalently,* ***(MB–RC1)*** *fails for H (Definition 13.5).*

A finite witness for OG4(H,w) is simply a directed cycle in $\mathcal{A}_{\neg w}(H)$. Likewise, a finite witness for OG4$^{\mathrm{MB}}(H)$ is a directed cycle in $\mathsf{A}(H,\mathcal{B}(H))$. Both witnesses are produced by standard cycle-detection on a finite directed graph.

Definition 16.3 (OG4 subtypes (operational))**.** *The OG4 witness is recorded in one of the following operational subtypes.*

- ***OG4b (true avoidance):*** *the directed cycle lives on the declared active face and avoids* every *block in $\mathcal{B}(H)$.*
- ***OG4a (face-drop detected):*** *the avoidance cycle is supported on a proper invariant subface of $S(H)$, indicating that the declared face is not minimal; in this case the pipeline refines the face and re-runs the Doeblin/recurrence search.*

Only OG4b is a terminal obstruction; OG4a triggers a deterministic reclassification step.

16.2 Why OG4 is a genuine obstruction

Proposition 16.4 (OG4 produces a tight minimizing trajectory with no canonical Doeblin blocks)**.** *If OG4$^{\mathrm{MB}}(H)$ holds, then there exists a periodic tight path π supported in H that avoids every word in $\mathcal{B}(H)$. In particular, no theorem can guarantee a bounded-gap supply of canonical Doeblin blocks along* all *tight minimizers in H.*

Proof. By Definition 16.2, $\mathsf{A}(H,\mathcal{B}(H))$ contains a directed cycle. As in Lemma 14.3, such a cycle converts into an explicit periodic tight path in H whose lifted dictionary state never enters an accepting configuration, hence it avoids every pattern in $\mathcal{B}(H)$. Since the path is tight, Lemma 4.7 shows that it has asymptotic mean cost λ_*, so it is a minimizing trajectory, yet it contains no occurrences of any block in $\mathcal{B}(H)$. ☐

Remark 16.5 (Relationship to OG1–OG3)**.** *OG4 can occur even when OG1–OG3 do not: an aperiodic tight SCC may exist, the face map may be consistent, and a strictly positive Doeblin block may be certified, but the canonical Doeblin family $\mathcal{B}(H)$ may still be avoidable by some tight trajectories in H. Accordingly, OG4 is the correct finite obstruction to record whenever* ***(MB–RC1)*** *fails for the canonical family.*

17 Contraction from a Certified Doeblin Word: the Finite Recurrence Branch

Once the finite obstruction taxonomy produces a certified Doeblin word on an aperiodic tight SCC, there is only one further way for "Doeblin ⇒ contraction" to fail on *all* tight trajectories: the Doeblin word might be avoidable. This is exactly OG4, witnessed by a directed cycle in the avoidance graph. When OG4 does not occur, the Doeblin word is *unavoidable* on the induced subshift over the SCC, hence appears with bounded gaps on every tight path, and uniform projective contraction follows with fully explicit constants.

We also record a second, quantitative route: along a general minimizing path, bounded gaps need not hold globally because tightness holds only in density. Nevertheless, if the minimizer spends positive lower density of tight time in the aperiodic SCC, then unavoidable recurrence forces a uniform positive lower density of Doeblin blocks, which is enough for an explicit exponential contraction rate.

17.1 Hilbert metric and Birkhoff contraction

17.2 Multi-block contraction: canonical families

The single-word recurrence gate RC1 is a special case of a stronger finite-state condition based on a canonical family of certified Doeblin blocks. In multi-block mode, one certifies that *some* block from the canonical family must occur with bounded gaps (MB–RC1), and the resulting contraction rate is uniform because the certificate records a uniform Doeblin margin across the family.

Proposition 17.1 (Unavoidable contraction blocks from MB–RC1)**.** *Let H be a tight aperiodic SCC with active face $S(H)$. Assume a canonical Doeblin family $\mathcal{B}(H)$ is certified (Section 13) and that **(MB–RC1)** holds for H (Definition 13.5). Let $\delta_{\min}$ denote the certified uniform Doeblin margin across the family (the field `delta_min_family` in the exported certificate). Then there exist explicit constants $L_{\text{syn}} = L_{\text{syn}}(H, \mathcal{B}(H))$ and $\tau = \tau(\delta_{\min}) \in (0, 1)$ such that every tight trajectory in H contains a block $b \in \mathcal{B}(H)$ within every window of length L_{syn}, and every such block occurrence contracts the Hilbert metric by factor at most τ. Consequently, every tight trajectory in H contracts at an explicit exponential rate depending only on (L_{syn}, τ).*

Proof. By Lemma 13.6, acyclicity of the multi-block avoidance graph implies the bounded-gap block-hit modulus L_{syn}. For each certified Doeblin block $b \in \mathcal{B}(H)$, the certificate's Doeblin margin yields a Birkhoff contraction factor $\tau_b < 1$ for the associated product on the active face. Using the uniform margin $\delta_{\min}$, one may take $\tau := \max_{b \in \mathcal{B}(H)} \tau_b = \tau(\delta_{\min})$. Composing contractions along successive block hits yields an exponential decay bound with rate constant proportional to $(-\log \tau)/L_{\text{syn}}$. □

Definition 17.2 (Hilbert (projective) metric)**.** *For $x, y \in \mathbb{R}^S_{>0}$, define*

$$d_H(x, y) := \log\Big(\max_{i \in S} \frac{x_i}{y_i}\Big) + \log\Big(\max_{i \in S} \frac{y_i}{x_i}\Big) = \log\Big(\max_{i,j \in S} \frac{x_i y_j}{x_j y_i}\Big).$$

This metric is invariant under positive scalar rescaling: $d_H(cx, dy) = d_H(x, y)$ for all $c, d > 0$.

Lemma 17.3 (Birkhoff contraction from entry bounds)**.** *Let $A \in \mathbb{R}^{S \times S}_{>0}$ satisfy $\min_{i,j \in S} A_{ij} \geq \varepsilon$ and $\max_{i,j \in S} A_{ij} \leq U$. Set*

$$\Delta := 2 \log \frac{U}{\varepsilon}, \qquad \tau := \tanh\Big(\frac{\Delta}{4}\Big) = \tanh\Big(\frac{1}{2} \log \frac{U}{\varepsilon}\Big) \in (0, 1).$$

Then for all $x, y \in \mathbb{R}^S_{>0}$,

$$d_H(Ax, Ay) \le \tau\, d_H(x, y).$$

Remark 17.4. *The constant τ is the standard Birkhoff contraction coefficient corresponding to the projective diameter bound Δ induced by entry ratio control; see Birkhoff [10] and modern treatments such as Bushell [11] or Lemmens–Nussbaum [12]. The certificate pipeline will provide explicit values of (ε, U) for Doeblin words.*

Definition 17.5 (Recurrence input for a Doeblin word)**.** *Fix an infinite tight path π in a tight SCC H and a Doeblin word w of length $|w|$ on H. Let $N_w(n;\pi)$ denote the number of occurrences of w starting among the first n edges of π. Let $N^{\mathrm{dis}}_w(n;\pi)$ denote the maximal number of pairwise disjoint occurrences of w whose starting indices lie in $\{0,\dots,n-1\}$; equivalently, the greedy selection rule yields $N^{\mathrm{dis}}_w(n;\pi) \ge \lfloor N_w(n;\pi)/|w| \rfloor$. We say that π satisfies one of the following recurrence inputs.*

- ***(RH1) Syndetic recurrence:*** *there exists $L_0 \in \mathbb{N}$ such that every length-L_0 subword of π contains an occurrence of w.*
- ***(RH2) Positive lower density:*** *there exists $\rho > 0$ such that $\liminf_{n\to\infty} \frac{N_w(n;\pi)}{n} \ge \rho$.*

Corollary 17.6 (Exponential projective contraction from Doeblin blocks)**.** *Let w be a Doeblin word on a face S whose face-restricted product $B(w)|_S$ has entry bounds (ε, U). Let $\tau \in (0,1)$ be the corresponding Birkhoff coefficient from Lemma 17.3. Then along any tight path π in H, the Hilbert distance between two positive face vectors transported by the tight product contracts exponentially provided π satisfies either recurrence input from Definition 17.5. More precisely:*

- *Under **(RH1)** with modulus L_0, one has $d_H(x_n, y_n) \le \tau^{\lfloor n/L_0 \rfloor}\, d_H(x_0, y_0)$ and hence $d_H(x_n, y_n) \le Ce^{-\kappa n} d_H(x_0, y_0)$ with $\kappa := \frac{1}{L_0}(-\log\tau)$ and $C := \tau^{-1}$.*
- *Under **(RH2)** with density ρ, let $N^{\mathrm{dis}}_w(n;\pi)$ be as in Definition 17.5. Then $d_H(x_n, y_n) \le \tau^{N^{\mathrm{dis}}_w(n;\pi)} d_H(x_0, y_0)$. Moreover, extracting disjoint occurrences by the greedy rule gives $N^{\mathrm{dis}}_w(n;\pi) \ge \lfloor N_w(n;\pi)/|w| \rfloor$, so the density bound implies an explicit exponential form $d_H(x_n, y_n) \le Ce^{-\kappa n} d_H(x_0, y_0)$ with $\kappa := \frac{\rho}{|w|}(-\log\tau)$ and $C := \tau^{-1}$.*

17.3 A tight-path version (uniform on the induced subshift over the tight SCC)

Theorem 17.7 (Finite recurrence dichotomy and contraction on tight paths)**.** *Fix an aperiodic tight SCC H of $G^{(0)}$ and assume OG2 and OG3 do not occur on H, so that a consistent active face $S(H)$ exists and an explicit Doeblin word w supported in H is certified with $B(w)$ strictly positive on $S(H) \times S(H)$.*

Then exactly one of the following holds.

(OG4) The avoidance graph $\mathcal{A}_{\neg w}(H)$ contains a directed cycle. Equivalently, there exists a periodic tight path in H that avoids w entirely (Proposition 16.4).

(UC) *The avoidance graph $\mathcal{A}_{\neg w}(H)$ is acyclic. Then every infinite tight path in H contains w with bounded gaps, with explicit gap modulus*

$$L_{\mathrm{syn}} := L_{\mathrm{syn}}(H, w) \le |V(H)| \cdot |w|$$

(Lemma 14.6). Consequently, for every tight path π supported in H and all $x, y \in \mathbb{R}^{S(H)}_{>0}$,

$$d_H\big(A^{(n)}(\pi)x,\, A^{(n)}(\pi)y\big) \;\le\; C\, e^{-\kappa n}\, d_H(x, y) \qquad (n \ge 1),$$

with explicit constants

$$\begin{aligned} &\varepsilon := m_*^{|w|}, && U := (dM_*)^{|w|}, \\ &\tau := \tanh\Big(\tfrac{1}{2}\log(U/\varepsilon)\Big) \in (0,1), && C := \tau^{-1}, \\ &\kappa := \frac{-\log\tau}{L_{\mathrm{syn}} + |w|}. \end{aligned}$$

Proof. If $\mathcal{A}_{\neg w}(H)$ contains a directed cycle, OG4 holds by Definition 16.1. If $\mathcal{A}_{\neg w}(H)$ is acyclic, Lemma 14.6 yields syndetic recurrence of w on *every* infinite path in H with gap bound L_{syn}. Since $B(w)$ is strictly positive on $S(H) \times S(H)$, the FTG bounds imply the window $\varepsilon \le B(w)_{ij} \le U$ on $S(H) \times S(H)$, hence the Birkhoff/Hilbert contraction factor is at most τ. Syndeticity implies at least $\lfloor n/(L_{\mathrm{syn}} + |w|)\rfloor$ contraction events by time n, so $d_H(\cdot) \le \tau^{\lfloor n/(L_{\mathrm{syn}}+|w|)\rfloor} d_H(x,y)$, and the exponential form follows as in Corollary 17.6. □

17.4 A mean-minimizer version (forced frequency from positive core time)

The preceding theorem is uniform on the induced subshift over H. For a general minimizing path, tightness holds only in density, so bounded gaps for w need not hold on the full path. However, if a minimizing path spends positive lower density of (tight) time in the aperiodic SCC H, then the finite recurrence modulus L_{syn} forces *positive lower density* of Doeblin blocks along that minimizer, which is sufficient for an explicit exponential contraction rate.

Theorem 17.8 (Contraction along minimizers in the Doeblin regime)**.** *Fix an aperiodic tight SCC H and a certified Doeblin word w on $S(H)$ as in Theorem 17.7. Assume OG4 does not occur for (H,w), so $\mathcal{A}_{\neg w}(H)$ is acyclic and $L_{\mathrm{syn}}(H,w)$ is defined. Let $L_0 := L_{\mathrm{syn}}(H,w) + |w|$.*

Let π be any one-sided minimizing path such that

$$\theta := \liminf_{n\to\infty} \frac{T_H(n;\pi)}{n} \; > \; 0,$$

where $T_H(n;\pi)$ counts times $t < n$ for which e_t is tight and its tail vertex lies in H (cf. Proposition 2.5). Then the Doeblin word w occurs along π with uniform positive lower density

$$\rho \;:=\; \liminf_{n\to\infty} \frac{N_w(n;\pi)}{n} \;\ge\; \frac{\theta}{L_0}$$

(Corollary 10.5). Consequently, for all $x, y \in \mathbb{R}^{S(H)}_{>0}$,

$$d_H\big(A^{(n)}(\pi)x,\, A^{(n)}(\pi)y\big) \;\le\; C\, e^{-\kappa n}\, d_H(x,y) \qquad (n \ge 1),$$

with explicit constants

$$\begin{aligned} &\varepsilon := m_*^{|w|}, && U := (dM_*)^{|w|}, \\ &\tau := \tanh\Big(\tfrac{1}{2}\log(U/\varepsilon)\Big), && C := \tau^{-1}, \\ &\kappa := \frac{\rho}{|w|}(-\log\tau) \;\ge\; \frac{\theta}{L_0|w|}(-\log\tau). \end{aligned}$$

Proof. Because OG4 does not occur, **(RC1)** holds for (H,w) and Corollary 10.5 yields $\rho \ge \theta/L_0$. Positive lower density of Doeblin blocks is precisely recurrence input (RH2) from Definition 17.5. Apply Corollary 17.6 with this ρ. □

Remark 17.9 (What is "gap-free" here)**.** *Theorems 17.7 and 17.8 remove recurrence as a hypothesis: the only alternative to contraction is the explicit finite-witness obstruction OG4. For minimizing paths, the positive-density condition $\theta > 0$ is guaranteed whenever an aperiodic tight SCC is selected in the tight-core dichotomy (Proposition 2.5).*

18 Rate Extraction from Doeblin and Finite Recurrence Gates

This section is a clean "rate pack" that converts the forcing outcomes (Sections 8, 9, and 10) and the avoidance certificate **(RC1)** into explicit exponential contraction constants. It is designed to be referenced directly by the pipeline contract in Section 20.

18.1 Constants from a Doeblin witness

Fix an aperiodic tight SCC H of $G^{(0)}$ with a consistent active face $S(H)$ and a certified Doeblin word w supported in H. Write $L := |w|$ and let $B(w)$ be the block product. By certification, $B(w)$ is strictly positive on $S(H) \times S(H)$.

Define the explicit FTG window and Birkhoff factor

$$\varepsilon := m_*^L, \qquad U := (dM_*)^L, \qquad \tau := \tanh\Big(\tfrac{1}{2}\log(U/\varepsilon)\Big) \in (0,1). \tag{17}$$

Then every occurrence of w contracts Hilbert distance on the active face by factor at most τ (Lemma 17.3 and Corollary 17.6).

18.2 Rate from syndetic recurrence (RC1 on a tight tail)

Proposition 18.1 (Rate from bounded gaps)**.** *Assume OG4 fails for (H, w), so the avoidance graph $\mathcal{A}_{\neg w}(H)$ is acyclic and the explicit gap modulus $L_{\mathrm{syn}} := L_{\mathrm{syn}}(H, w)$ is defined. Let π be any infinite path whose tail is a path in H (in particular, any tight minimizer supported in H). Then the occurrences of w along π are syndetic with gap bound L_{syn}. Consequently, for all $x, y \in \mathbb{R}^{S(H)}_{>0}$ and all $n \geq 1$,*

$$d_H\big(A^{(n)}(\pi)x,\ A^{(n)}(\pi)y\big) \ \leq\ C\, e^{-\kappa n}\, d_H(x,y)$$

with

$$C := \tau^{-1}, \qquad \kappa := \frac{-\log\tau}{L_{\mathrm{syn}} + L}. \tag{18}$$

Proof. If $\mathcal{A}_{\neg w}(H)$ is acyclic, Lemma 14.6 provides syndetic recurrence on every infinite path in H. The contraction estimate is exactly the (RH1) branch of Corollary 17.6, using (17). □

18.3 Rate from forced positive density (mean-minimizers with zero slack density)

Proposition 18.2 (Rate from forced word density)**.** *Assume OG4 fails for (H, w), and let $L_0 := L_{\mathrm{syn}}(H, w) + L$. Let π be any one-sided path and define*

$$\theta := \liminf_{n\to\infty} \frac{T_H(n;\pi)}{n}, \qquad \eta := \limsup_{n\to\infty} \frac{\mathrm{SlackMass}_n(\pi)}{n}.$$

If $\theta > 0$ and $\eta = 0$, then

$$\rho := \liminf_{n\to\infty} \frac{N_w(n;\pi)}{n} \ \geq\ \frac{\theta}{L_0} \tag{19}$$

(Corollary 10.5). Consequently, for all $x, y \in \mathbb{R}^{S(H)}_{>0}$ and all $n \geq 1$,

$$d_H\big(A^{(n)}(\pi)x,\ A^{(n)}(\pi)y\big)\ \leq\ C\,e^{-\kappa n}\,d_H(x,y)$$

with

$$C := \tau^{-1}, \qquad \kappa := \frac{\rho}{L}(-\log\tau)\ \geq\ \frac{\theta}{L_0 L}(-\log\tau). \tag{20}$$

Proof. Equation (19) is Corollary 10.5. Given a lower density ρ of starting indices, extract a disjoint subfamily of occurrences by the standard greedy rule. Among the first n steps this produces at least $\lfloor \rho n/L \rfloor$ disjoint w-blocks. Each disjoint block contracts by factor at most τ, yielding $d_H(\cdot) \leq \tau^{\lfloor \rho n/L \rfloor} d_H(x,y) \leq \tau^{-1} e^{-(\rho/L)(-\log\tau)n} d_H(x,y)$, which is (20). ☐

Remark 18.3 (Where θ enters)**.** *The parameter θ measures the asymptotic lower density of tight time spent in the chosen aperiodic SCC H. For tight minimizers supported in H one has $\theta = 1$, so (20) reduces to an explicit path-independent lower bound. For general minimizers, θ is recorded as a trajectory-level input (or estimated in applications) while the constants τ and L_0 are fully witness-derived.*

18.4 Multi-block rate pack (canonical families)

In canonical multi-block mode, contraction events are the occurrences of *any* certified block from the canonical family $\mathcal{B}(H)$. The certificate provides a uniform Doeblin margin $\delta_{\min}$ across the family and a syndetic modulus $L_{\mathrm{syn}}(H, \mathcal{B}(H))$ when MB–RC1 holds. This yields a uniform contraction rate using the worst-case block contraction factor and the maximal block length.

Proposition 18.4 (Rate from bounded-gap multi-block hits)**.** *Assume MB–RC1 holds for a tight aperiodic SCC H with canonical family $\mathcal{B}(H)$. Let $L_{\mathrm{syn}} := L_{\mathrm{syn}}(H, \mathcal{B}(H))$ and let $L_{\max} := \max_{b\in\mathcal{B}(H)} |b|$. Let $\tau = \tau(\delta_{\min})$ be the uniform contraction factor associated to the certified uniform Doeblin margin across the family. Then along every tight trajectory in H, the Hilbert metric contracts exponentially with rate*

$$d_H(x_n, y_n)\ \leq\ \tau^{-1}\exp\Big(-\frac{-\log\tau}{L_{\mathrm{syn}} + L_{\max}}\,n\Big)\,d_H(x_0, y_0).$$

Proof. MB–RC1 implies at least one family block begins within every window of length L_{syn}. Select a disjoint subfamily of block occurrences by the greedy rule that keeps an occurrence and then skips forward by $L_{\max}$ steps. Over n steps this yields at least $\lfloor n/(L_{\mathrm{syn}} + L_{\max}) \rfloor$ disjoint contraction blocks. Each block contracts by factor at most τ, giving the stated bound. ☐

Proposition 18.5 (Rate from positive density of tight time in multi-block mode)**.** *Suppose a minimizing trajectory spends lower density at least $\theta > 0$ of time on tight edges in an aperiodic SCC H. If MB–RC1 holds for H with modulus L_{syn} and maximal block length $L_{\max}$, then the Hilbert metric contracts at rate at least*

$$\kappa_{\mathrm{MB}}\ :=\ \frac{\theta}{L_{\mathrm{syn}} + L_{\max}}\big(-\log\tau\big),$$

where $\tau = \tau(\delta_{\min})$ is the uniform contraction factor from the certified uniform Doeblin margin across the family.

Proof. On the subsequence of indices corresponding to tight time spent in H, apply Proposition 18.4 with n replaced by the tight-time count. The lower density θ converts the contraction rate in tight time to a global-in-n rate by standard density comparison. ☐

19 Complete Tight-Core Boundary Theorem

This section rewrites the flagship boundary classification so it is finite and witness-producing. The theorem stops at the boundary objects that are decidable from finite data: tight-core type, face-map consistency, and Doeblin-word existence.

Recurrence along a chosen minimizing trajectory is a separate layer and is used only under explicit recurrence inputs.

Definition 19.1 (Obstruction gates (OG1–OG4))**.** *The finite boundary analysis is organized around four obstruction gates.*

- ***OG1:*** *every tight SCC is imprimitive (*$\mathrm{per}(H) > 1$*), so no aperiodic tight core exists.*
- ***OG2:*** *an aperiodic tight SCC exists, but active-face consistency fails (Definition 11.1), producing a cycle witness (Lemma 11.9).*
- ***OG3:*** *a consistent active face exists, but there is no Doeblin word on that face; this yields a finite cut or product-automaton non-reachability witness (Algorithm 6).*
- ***OG4:*** *a Doeblin word exists, but it is not recurrent along the tight dynamics; this yields an explicit avoidance-cycle witness (Lemma 14.3) and is formalized as Definition 16.1.*

Corollary 19.2 (Obstruction witnesses are explicit)**.** *If any obstruction gate from Definition 19.1 holds, then the certificate pipeline produces a finite, checkable witness object (cycle, cut, automaton non-reachability, or avoidance cycle). In particular, in the presence of an obstruction gate, no gap-free Doeblin + recurrence contraction conclusion can be asserted on the corresponding tight core.*

Theorem 19.3 (Complete Tight-Core Boundary Theorem (finite, witness-producing form))**.** *Let $G^{(0)}$ be the tight graph associated to a calibrated potential (Definition 2.2). Then exactly one of the following holds.*

e-only tight core. *Every SCC H of $G^{(0)}$ is imprimitive, i.e.* $\mathrm{per}(H) > 1$ *(Definition 2.3). A finite witness is the SCC decomposition of $G^{(0)}$ together with the computed periods* $\mathrm{per}(H)$.

tight SCC exists. *There exists an aperiodic SCC H of $G^{(0)}$, i.e.* $\mathrm{per}(H) = 1$*. Fix such an H. Then exactly one of the following holds on H.*

face instability on H. *The descriptor reduction fails to provide a consistent face map on H (Definition 11.1). A finite witness is a directed cycle word in H with marked indices (i, j) witnessing spurious activation, as in Lemma 11.9.*

the active face of H. *A consistent face map exists on H with active face $S(H)$, but there is no word w supported in H whose face-restricted block product is strictly positive on $S(H) \times S(H)$. A finite witness is either*

- *a cut witness showing $\Gamma(H)$ is not strongly connected, or*
- *a Boolean-pattern product-automaton non-reachability certificate for the all-ones pattern state $(v_0, \mathbf{1})$ produced by Algorithm 6.*

ord is certified on H. *There exists a word w supported in H such that the face-restricted block product $B(w)$ is strictly positive on $S(H) \times S(H)$. Equivalently, the Boolean-pattern product-automaton test reaches a return state $(v_0, \mathbf{1})$ and returns an explicit witness word w (Corollary 11.6). The witness object is the triple $(H, S(H), w)$.*

Moreover, if a minimizing path π admits an aperiodic tight SCC in its tight-core dichotomy (Proposition 2.5), then applying the aperiodic branch above to that SCC returns a finite boundary object: an explicit OG2 witness, an explicit OG3 witness, or an explicit Doeblin word witness.

Proof map. Compute the SCC decomposition of $G^{(0)}$ and the period $\mathrm{per}(H)$ of each SCC (Definition 2.3). If all SCCs satisfy $\mathrm{per}(H) > 1$, we are in (OG1). Otherwise, fix an aperiodic SCC H.

Face-map consistency on H is a finite check on the template sparsity patterns and yields an explicit inconsistency cycle witness when it fails (Lemma 11.9). Assuming a consistent face map, Doeblin-word existence on $S(H)$ is equivalent to reachability of a return state $(v_0, \mathbf{1})$ in the Boolean-pattern product automaton. Algorithm 6 returns either an explicit witness word w or a finite non-reachability certificate (Corollary 11.6). These outcomes are mutually exclusive and exhaustive, yielding (OG2), (OG3), or (**Doeblin**).

Finally, Proposition 2.5 identifies when a minimizing path supplies an aperiodic tight SCC to which the aperiodic branch applies. □

20 Pipeline Contract (recurrence derived, finite-witness branches)

This section states the end-to-end contract of the gap-free pipeline once the finite recurrence layer and the rate extraction layer are included. The key upgrade is that *recurrence is no longer an external hypothesis*. Instead, the pipeline returns either a *finite obstruction witness* (OG1–OG4) or an *explicit contraction certificate with rate*.

20.1 Inputs and canonical objects

The pipeline assumes the standard reduced-graph data already used by the obstruction checks:

- a finite reduced directed graph $G = (V, E)$,
- for each edge $e : u \to v$ a nonnegative template (or fixed-length block) $T_e \in \mathbb{R}_+^{d\times d}$,
- the calibrated potential ϕ and reduced costs $c_\phi(e) \ge 0$ (so the tight edge set $E^{(0)} = \{c_\phi = 0\}$ is defined),
- the canonical support sets $S(v) \subseteq \{1, \dots, d\}$ per descriptor state.

From these, the pipeline constructs:

- the tight graph $G^{(0)} = (V^{(0)}, E^{(0)})$,
- SCC decomposition and periods $\mathrm{per}(H)$,
- face-map validation and SCC face $S(H) = \bigcap_{v\in H} S(v)$ when consistent (Definition 11.1),
- the coordinate digraph $\Gamma(H)$ on $S(H)$ (Definition 11.4),
- the Boolean-pattern product automaton for Doeblin-word search (Algorithm 6),
- for a returned Doeblin word w, the avoidance graph $\mathcal{A}_{\neg w}(H)$ and (if acyclic) the modulus $L_{\mathrm{syn}}(H, w)$ (Algorithm 7).

20.2 Branch logic and output artifacts

Frontier localization. Every output branch can be augmented with a *frontier report* (Section 21) recording the minimal failing gate and the implicated boundary loci. In obstruction branches, the returned artifact is a structured `obstruction_witness` object (Section 23); in the certificate branch, the returned artifact is a `robust_certificate` object (Section 25).

Both object types may additionally export a `local_invariance` radius whenever one is certified. In the YES case, this is the robust radius $\varepsilon^\star$. In the NO case, under support-preserving perturbations the obstruction witness can export the radius from Theorem 29.1. In particular, on fixed-support template families (Definition 3.1) this condition is part of the model contract, so the NO-side invariance radius is certified whenever the cost-side tight-core radius is certified.

The pipeline returns exactly one of the following mutually exclusive outcomes.

(OG1 witness) No aperiodic SCC exists in $G^{(0)}$. Return the SCC/period table certifying that every tight SCC is imprimitive.

Optional period-blocking upgrade (imprimitive tight cores). When OG1 occurs, one may optionally apply a finite preprocessing step. For each tight SCC H of period $p = \mathrm{per}(H) > 1$, build the p-step return graph $H^{[p]}$ whose edges correspond to tight length-p paths in H, labeled by the product block of templates along the path. Then rerun the pipeline on $H^{[p]}$. Any contraction certificate on $H^{[p]}$ lifts to a contraction statement on H with the same Hilbert contraction factor per block and with an exponential rate rescaled by $1/p$. Accordingly, OG1 should be read as "no aperiodic tight SCC at step 1"; the optional blocking reduction extends the decidable island to imprimitive tight cores while preserving finite witnesses.

(OG2 witness) An aperiodic tight SCC H exists, but the proposed face sets fail to form a face map on H. Return the inconsistency cycle witness from Lemma 11.9.

(OG3 witness) An aperiodic tight SCC H with a valid face map exists, but no Doeblin word exists on $S(H)$. Return either:

- a cut witness $(H, S(H), A, B)$ from Lemma 11.10, or
- a product-automaton non-reachability certificate (Corollary 11.6).

(OG4 witness) A Doeblin word w on $S(H)$ is certified, but RC1 fails for (H, w). Return a directed cycle in $\mathcal{A}_{\neg w}(H)$, which yields an explicit periodic tight minimizer avoiding w (Proposition 16.4).

action certificate) A Doeblin word w on $S(H)$ is certified and RC1 holds for (H, w). Return:

- the witness word w and the modulus $L_{\mathrm{syn}}(H, w)$,
- the FTG window parameters (ε, U) for the Doeblin block on $S(H)$,
- the Hilbert contraction factor τ and the exponential rate parameters (C, κ) as in Theorem 17.7.

This certificate implies uniform exponential projective contraction along every tight minimizer supported in H.

20.3 How general minimizers are handled

For a general one-sided minimizing path π, the tight edges need not appear with bounded gaps. The pipeline therefore records the correct forcing upgrade:

- if π has nontrivial tight time in the chosen SCC H (measured by $\theta = \liminf T_H(n;\pi)/n > 0$) and slack density 0, then the forcing lemma yields a *positive lower density* of Doeblin blocks (Corollary 10.5),
- this density supplies the recurrence input needed by the contraction module, yielding an explicit rate (Theorem 17.8 and the formulas packaged in Section 18).

Accordingly, the pipeline contract records two rate modes:

- a *tight-minimizer* rate (C, κ) that depends only on finite witnesses (H, w) and the FTG bounds,
- a *mean-minimizer* rate that additionally depends on the measurable parameter θ of the specific trajectory.

Remark 20.1 (What is now fully finite)**.** *All branching decisions up through OG4 and the tight-minimizer contraction certificate are entirely finite and witness-producing. The only non-finite input that can remain in applications is an estimate of how much time a given minimizer spends in the SCC H (the parameter θ). This is the correct remaining interface: it is a property of the chosen trajectory, not a hidden structural assumption.*

21 Frontier report artifact (maximal failure localization)

Every pipeline outcome can be strengthened to a *frontier report*. The idea is to replace a bare YES/NO with a minimal-gate diagnosis and a quantitative proximity-to-boundary summary. This is the mechanism that turns the decidable island into a *decidability frontier*: when certification fails, we return *where* it fails (OG1–OG4), *why* it fails (the witness), and *which boundary locus* is implicated (margin-collapse or combinatorial failure).

21.1 Object-level contract

Definition 21.1 (Frontier report object)**.** *A* frontier report *is a structured object containing:*

- *an* `outcome_label` *in* $\{$`contraction,OG1,OG2,OG3,OG4`$\}$*;*
- *a* `minimal_gate` *equal to the first failing predicate in the branch logic (or* `contraction` *in the YES case);*
- *a* `critical_margins` *payload recording the decisive margins (typically γ and δ when defined, and the robust radius ε^* when available);*
- *an optional* `local_invariance` *payload recording a certified neighborhood on which the* `minimal_gate` *label is provably constant (when such a radius is available from Tight-Core Stability and support stability);*

- *a list `frontier_loci` indicating which canonical boundary loci are relevant for the outcome (e.g. $\gamma \to 0$, $\delta \to 0$, recurrence-gate failure (RC1 failure in single-block mode, or MB–RC1 failure in canonical multi-block mode));*

- *an optional `witness_ref` pointing to the corresponding obstruction witness file (in the NO case).*

Lemma 21.2 (Minimal-gate uniqueness and local invariance)**.** *For any fixed certified instance, the `minimal_gate` field in Definition 21.1 is well-defined and unique. Moreover, suppose the instance comes equipped with a tight-core stability radius $\varepsilon_{\text{tight}} > 0$ (Section 26). Then the `minimal_gate` label is locally invariant under any perturbation of size at most $\varepsilon_{\text{tight}}$ that preserves the Boolean support patterns of the templates on the relevant core face blocks (for instance by the support-stability threshold of Section 25.2). In particular, in the YES case the robust radius $\varepsilon^\star$ provided by a contraction certificate implies invariance of `minimal_gate` throughout the $\varepsilon^\star$-neighborhood of the instance.*

Proof. Uniqueness follows because the branch logic orders the predicates OG1–OG4 linearly and `minimal_gate` is defined as the first failing predicate.

For local invariance, tight-core stability keeps the tight edge set and tight SCC structure unchanged on perturbations of size at most $\varepsilon_{\text{tight}}$. Under support-pattern preservation, the face-map validity predicate, the Doeblin reachability predicate on the Boolean-pattern automaton, and the avoidance-cycle predicate on the avoidance graph are unchanged, since these are purely combinatorial predicates on the same finite graphs. Hence the first failing predicate is unchanged, so `minimal_gate` is invariant on that neighborhood. In the YES case, the certificate-provided radius $\varepsilon^\star$ is chosen no larger than the tight-core and support-pattern stability radii, so the same conclusion holds on the $\varepsilon^\star$-ball. □

21.2 Self-verifying decision objects

A frontier report is intended to be *more than a label*: it should be a compact object that an independent reader can check using only finite computations on the stored witnesses.

Lemma 21.3 (Frontier reports are finitely verifiable)**.** *Given a frontier report (Definition 21.1) together with its referenced witness payload (a `robust_certificate` in the `contraction` case, or an `obstruction_witness` in the OG cases), there is a finite verification procedure that checks:*

- *the `minimal_gate` claim (the witness satisfies the corresponding OG predicate, and all earlier predicates fail);*

- *the validity of the stored `local_invariance` radius when `local_invariance.certified=true`;*

- *in the `contraction` case, the Doeblin and recurrence certificates and the derived rate pack.*

In particular, inside the certified class the frontier report can be treated as a self-contained decision certificate rather than a narrative summary.

Proof. Each gate OG1–OG4 is a predicate on finite graphs/automata with explicit stored witnesses (Section 23), and the contraction certificate is likewise finite (Section 25). The ordered-branch logic defines `minimal_gate` as the first failing predicate, so verifying it amounts to checking the witness predicate and checking the negation of all earlier predicates. Local invariance reduces to checking that the stored radius is bounded by the relevant stability radii exported by Tight-Core Stability and support stability, and then appealing to Theorem 29.1 in the NO case or the robustness contract in the YES case. All steps are finite. □

Algorithm 8 VERIFYFRONTIERREPORT (finite witness check)

Require: Frontier report R and referenced payload object W

Ensure: PASS/FAIL with a list of failed checks

Check that R.`outcome_label` = R.`minimal_gate`; else FAIL

if R.`minimal_gate` = `contraction` **then**

  Verify W is a valid `robust_certificate` and contains $(H, S(H), w)$ (single-block) or $(H, S(H), \mathcal{B}(H))$ (multi-block), the corresponding recurrence-gate modulus, and the rate pack

  Recompute (or re-check) Doeblin positivity on $S(H)$ and the appropriate avoidance-acyclicity predicate (single-block $\mathsf{A}(H, w)$ or multi-block $\mathsf{A}(H, \mathcal{B}(H))$)

else

  Verify W is a valid `obstruction_witness` of type R.`minimal_gate`

  Check the stored witness predicate (period table / inconsistency cycle / nonreachability / avoidance cycle)

  Check that all earlier predicates fail on the same finite structures

end if

if R.`local_invariance.certified` = `true` **then**

  Verify the reported radius is bounded by the applicable stability thresholds (tight-core and support-stability)

end if

return PASS if no check fails; otherwise return FAIL with failures

A pipeline-facing JSON schema and an example object are included:

- `frontier_report_schema.json`

The frontier report is intended to be embedded inside both pointwise artifacts (robust certificates and obstruction witnesses) and region-mode reports.

22 Frontier atlas report artifact (region-level frontier compression)

Pointwise outcomes (Section 21) identify a minimal failing gate at a single parameter instance. Region-mode certification and global-on-Θ verification require a *region-level* object that compresses the full region report into a single, headline artifact: *which regimes occur on the region, which minimal gates are responsible, and where the boundary loci concentrate.*

The *frontier atlas report* is that compression. It is designed to be the primary human-facing object for **OP–FRONTIER–MAP** and **OP–ATLAS–FIN**:

- It merges chamber summaries, obstruction witnesses, and near-degeneracy localization into one object.
- It is witness-producing: every noncontraction label points to a concrete obstruction witness or a declared degeneracy neighborhood.
- It is stable under refinement: adding more sample points or shrinking the uncovered remainder can only refine the atlas, never invalidate it.

22.1 Object-level contract

Definition 22.1 (Frontier atlas report object)**.** *A* frontier atlas report *on a compact normalized family* Θ *is a structured object containing:*

- *a* `summary` *block stating whether* good-global contraction universality *holds, and if not, listing the minimal-gate types present;*
- *a list* `chambers` *of region pieces (chambers or certified balls) each labeled by an* `outcome_label` *in* {`contraction`,`OG1`,`OG2`,`OG3`,`OG4`,`near_degeneracy`}*, together with references to the supporting certificate or witness;*
- *for each chamber, a* `chamber_invariance` *block exporting a single certified decision radius lower bound* `eps_dec` *(uniform over the chamber's certified subcover), together with* `cover_refs` *from which* `eps_dec` *is derived;*
- *a* `coverage` *block stating whether the certified subcover closes on the whole region, and if not, localizing the uncovered remainder inside a declared neighborhood of named degeneracy loci;*
- *a* `stratification` *block recording the regime labels present, the chamber IDs grouped by label, and the list of boundary loci across which labels can change;*
- *a list* `loci` *of boundary/degeneracy loci relevant to the region, with explicit localization notes (for example: "contained in the declared* $\gamma \approx 0$ *neighborhood");*
- *an* `index` *of referenced certificate IDs and witness files so the atlas is self-contained as an audit artifact.*

Remark 22.2 (Chamber-level carry-forward radius)**.** *A chamber label is only useful operationally if it carries a* single *radius statement that survives downstream composition. For each chamber* Ω *the atlas exports* `chamber_invariance.eps_dec`*, defined as the minimum of the certified local invariance radii of the items listed in* `chamber_invariance.cover_refs`*. By construction,* Ω *is covered by these items (certified balls and/or witnesses), and every item in the subcover has outcome label equal to* `outcome_label`*. Therefore every point of* Ω *lies in a neighborhood on which the same minimal-gate label is invariant, and* `eps_dec` *is a uniform lower bound for that neighborhood size across the chamber.*

22.2 Stratification and completeness

The atlas is more than a summary. It is a finite, witness-producing *stratification* of the region by minimal-gate regime labels, together with a certified neighborhood size on which the label is invariant. The next theorem makes that meaning explicit.

Theorem 22.3 (Frontier stratification with chamberwise invariance radii)**.** *Let* Θ *be a compact normalized parameter family in the certified class, and assume the fixed-support contract holds on* Θ *(Definition 3.1). Run the global cover/atlas decider of Theorem 32.3 and compress its output by* CompressToFrontierAtlas*. Then the embedded* `FrontierAtlasReport` *has the following properties.*

- ***Finite regime partition.*** *The list* `chambers` *is finite and each chamber carries a unique* `outcome_label` *in* {`contraction`,`OG1`,`OG2`,`OG3`,`OG4`,`near_degeneracy`}*. Every noncontraction label is supported either by an explicit obstruction witness file or by a declared degeneracy neighborhood recorded in* `loci`*.*

- ***Local invariance with an exported radius.*** *For every chamber Ω the atlas exports a single certified decision invariance radius lower bound* `chamber_invariance.eps_dec`*. For each $\theta \in \Omega$ there exists a verified center item in* `cover_refs` *whose local invariance radius is at least* `eps_dec` *and whose minimal-gate label equals* `outcome_label`*. Consequently the regime label is constant on a neighborhood of size at least* `eps_dec` *around every point of Ω (in the declared perturbation model).*

- ***Completeness up to boundary loci.*** *If* `coverage.uncovered_is_empty` *is true, then good-global contraction universality holds on Θ and the atlas has only the label* `contraction`*. If* `coverage.uncovered_is_empty` *is false, then* `coverage.uncovered_loci` *lists the degeneracy types near which the uncovered remainder is certified to lie. In particular, outside the declared degeneracy neighborhood the atlas provides a finite regime stratification by invariant labels with explicit neighborhood radii.*

- ***Discontinuity confinement.*** *The only locations where the minimal-gate label can change under refinement or perturbation are contained in neighborhoods of the loci listed in* `stratification.discontinuity_l`

Proof. Termination and finiteness follow from the finite-state nature of every subroutine (tight-core extraction, product-automaton search, and avoidance-cycle detection) together with the compactness-driven region-mode refinement in Theorem 32.3. Each center item in the region-mode subcover exports a certified local invariance radius by Theorem 1.7. The compression algorithm defines `chamber_invariance.eps_dec` as the minimum local invariance radius among the items that cover the chamber, so the claimed uniform lower bound holds. When the uncovered remainder is empty the certified global case holds by definition; otherwise the decider localizes the uncovered remainder inside the declared degeneracy neighborhood, so the stratification is complete on the complement. Finally, label changes can only occur where either the tight core changes or a certificate margin collapses, which is precisely what the recorded loci represent under the fixed-support contract. □

The atlas object is intended to be derivable from a full `GlobalUniversalityReport` (Section 32.4) by discarding only bookkeeping. In particular, it retains exactly the data needed to answer: *"Where is the decidable island boundary on this region, and what finite witnesses certify that boundary?"*

A pipeline-facing JSON schema and example objects are included:

- `frontier_atlas_report_schema.json`
- `frontier_atlas_report_ThetaCan_certified.json`
- `frontier_atlas_report_ThetaCan_atlas.json`

The global report schema optionally embeds the frontier atlas report under the field `frontier_atlas`.

23 Obstruction witness artifact (OG1–OG4)

The pipeline is designed to be witness-producing. Accordingly, whenever certification fails, the output is not a null result but a finite witness object that can be checked independently.

23.1 Object-level contract

Definition 23.1 (Obstruction witness object)**.** *An* obstruction witness *is a structured object containing:*

- *a model fingerprint (`model_hash`);*
- *an `outcome_label` in {OG1,OG2,OG3,OG4} and a `minimal_gate` equal to that label;*
- *a `witness` payload whose required fields depend on the obstruction type:*
  - ***OG1:*** *a tight-SCC period table certifying that no step-1 aperiodic tight SCC exists; optional period-blocking hint.*
  - ***OG2:*** *a tight SCC id, an inconsistent edge, and a finite cycle word together with marked indices certifying spurious activation outside the face.*
  - ***OG3:*** *a tight SCC id and face support together with either a cut witness (A, B) or a finite non-reachability certificate in the Boolean-pattern product automaton.*
  - ***OG4:*** *a tight SCC id, the Doeblin word, and a directed cycle in the avoidance graph (plus an optional derived avoiding periodic word).*
- *an optional `margins_snapshot` section recording any computed margins relevant to frontier localization.*
- *a `local_invariance` payload recording whether a certified perturbation neighborhood is available on which the obstruction label is stable, together with the corresponding radius when certified.*
- *an optional `frontier_loci` list, indicating which canonical boundary loci the obstruction corresponds to.*

23.2 Concrete JSON schema

A pipeline-facing JSON schema is included:

- `obstruction_witness_schema.json`

The obstruction witness object is designed to be embedded as a `witness_ref` target inside a frontier report.

24 Schema: MatherSoficModel (OP–MATHER–EFF artifact)

This section records a lightweight export schema for the sofic extremal model of Definition 15.2. The object is designed to be self-contained: it describes a tight SCC shift X_H, the avoidance automaton for a chosen word w, and the finite witness concluding either RC1 (bounded-gap recurrence) or OG4 (an avoidance cycle).

Files. The JSON schema is `mather_sofic_model_schema.json`. An example payload is `mather_sofic_model_exa`

Core fields.

- `core`: basic finite data for the tight SCC shift X_H (counts and an explicit transition list), with optional `face_support`.
- `avoidance_automaton`: state and edge counts for the avoidance automaton $\mathsf{A}(H, w)$.
- `outcome`: either `RC1` (with `gap_bound_B`) or `OG4` (with an explicit `avoidance_cycle`).

Verification. A verifier checks:

- schema validity,
- that each listed transition is an edge of the declared SCC shift,
- if `RC1`: that the reported `gap_bound_B` is consistent with an acyclicity claim for the avoidance graph (when the full graph is provided by reference),
- if `OG4`: that `avoidance_cycle` forms a directed cycle in the avoidance graph.

In the full pipeline, the needed supporting graphs are already present in the certificate or obstruction witness payloads; the model object acts as a compact symbolic summary.

25 Robust Certificate Contract

This section freezes a single portable object that stores *all* margins and thresholds needed to make robustness claims mechanically traceable. Later sections may assert "the conclusion survives perturbation" only by citing explicit fields of this object.

25.1 Perturbation model

We allow two perturbation channels:

- **Cost perturbations.** The edge cost vector $c : E \to \mathbb{R}$ is perturbed to $\tilde{c} = c + \Delta c$ with

$$\|\Delta c\|_\infty := \max_{e \in E} |\Delta c(e)| \le \varepsilon_c.$$

- **Template perturbations.** Each template matrix $A^{(k)} \in \mathbb{R}_+^{d \times d}$ is perturbed to $\tilde{A}^{(k)} = A^{(k)} + \Delta A^{(k)}$ under one of the following metrics, which must be fixed in the certificate:
  - *Absolute entrywise:* $\|\Delta A\|_{\infty,\mathrm{ent}} := \max_{k,i,j} |\Delta A_{ij}^{(k)}| \le \varepsilon_A$.
  - *Relative entrywise:* $|\Delta A_{ij}^{(k)}| \le \varepsilon_A \, A_{ij}^{(k)}$ for all k, i, j.

When both channels are present, we write $\varepsilon := (\varepsilon_c, \varepsilon_A)$ and define the *robust radius* as the largest admissible pair $\varepsilon^\star$ stored in the robust certificate.

25.2 Support-pattern stability (support-preserving perturbations)

Several obstruction gates (OG2–OG4) and the Doeblin-word decision procedure depend only on *Boolean support patterns* of the templates on a fixed core/face block. To make "NO" outcomes as robust as "YES" outcomes, we include an explicit support-stability margin in the contract.

Definition 25.1 (Support-preserving perturbation model)**.** *A template perturbation model is* support-preserving *if every perturbation satisfies*

$$A_{ij}^{(k)} = 0 \implies \Delta A_{ij}^{(k)} = 0 \qquad \text{for all } k, i, j.$$

Equivalently, the sparsity pattern of each template is fixed and only the positive entries are perturbed.

Remark 25.2 (Why support-preserving is a natural contract)**.** *In most applications the sparsity pattern of each template is structural (forbidden couplings remain forbidden), and parameter families vary only the weights on the allowed entries. This is formalized as the fixed-support template-family contract in Definition 3.1. Under that contract, support-preserving perturbations are not an extra hypothesis: they are exactly the perturbations that stay inside the declared family.*

Definition 25.3 (Support margin and support-stability radius)**.** *Assume support-preserving perturbations. Define the* support margin

$$\eta := \min\{A_{ij}^{(k)} : A_{ij}^{(k)} > 0\}$$

computed over all template entries that are relevant for the chosen core/face blocks used by the pipeline. For absolute entrywise perturbations, define the support-stability threshold

$$\varepsilon_{\text{supp}} := \eta/2.$$

For relative entrywise perturbations, one may take $\varepsilon_{\text{supp}} := 1/2$ (any factor < 1 suffices). The certificate must store both η and the chosen threshold $\varepsilon_{\text{supp}}$.

Lemma 25.4 (Boolean support-pattern invariance)**.** *Assume support-preserving perturbations (Definition 25.1). If $\varepsilon_A < \varepsilon_{\text{supp}}$ from Definition 25.3, then for every template index k and every entry (i, j),*

$$A_{ij}^{(k)} > 0 \iff \tilde{A}_{ij}^{(k)} > 0,$$

so the Boolean support pattern used by the Doeblin-word automaton and the avoidance graph is unchanged throughout the ε_A-neighborhood.

Proof. If $A_{ij}^{(k)} = 0$ then support-preserving implies $\tilde{A}_{ij}^{(k)} = 0$. If $A_{ij}^{(k)} > 0$ then $A_{ij}^{(k)} \geq \eta$ and the perturbation bound gives $\tilde{A}_{ij}^{(k)} \geq A_{ij}^{(k)} - \varepsilon_A > \eta - \eta/2 = \eta/2 > 0$. □

25.3 Margins and openness predicates

Fix a calibrated subaction $u : V \to \mathbb{R}$ and level λ_*, and reduced costs c_u as in (2). Write $H := \{e \in E : c_u(e) = 0\}$ for the tight edge set.

Definition 25.5 (Tightness gap margin)**.** *Assume $H \neq E$. The* tightness gap *is*

$$\gamma := \min\{c_u(e) : e \in E \setminus H\}.$$

If $H = E$ we set $\gamma := +\infty$.

Lemma 25.6 (Openness of the tightness sign test). *Let $\tilde{c} = c + \Delta c$ with $\|\Delta c\|_\infty \le \varepsilon_c$ and keep (u, λ_*) fixed. Then for every edge $e \in E$,*

$$|\tilde{c}_u(e) - c_u(e)| \le \varepsilon_c, \qquad \tilde{c}_u(e) := \tilde{c}(e) - \lambda_* + u(\mathrm{tail}(e)) - u(\mathrm{head}(e)).$$

In particular, if $\varepsilon_c < \gamma/2$, then

$$e \in H \implies \tilde{c}_u(e) \le \gamma/2, \qquad e \notin H \implies \tilde{c}_u(e) \ge \gamma/2.$$

Lemma 25.6 is the primitive "openness" statement behind tight-core stability; Tight-Core Stability upgrades it to a combinatorial invariance theorem for the tight graph and its SCC decomposition.

Next, fix a Doeblin word witness $w = w_1 \cdots w_L$ (over the template index alphabet) and define its product

$$P_w := A^{(w_L)} \cdots A^{(w_1)}.$$

Let $\delta := \min_{i,j}(P_w)_{ij}$ be the Doeblin margin on the relevant block.

Definition 25.7 (Uniform entry bound). *Fix a bound $M \ge 1$ satisfying $\max_{k,i,j} A^{(k)}_{ij} \le M$ on the block where the Doeblin witness is evaluated. The certificate must store the value of M used in robustness inequalities.*

Lemma 25.8 (Openness of Doeblin positivity under absolute perturbations). *Assume absolute entrywise perturbations with $\|\Delta A\|_{\infty,\mathrm{ent}} \le \varepsilon_A$ and $\max_{k,i,j} A^{(k)}_{ij} \le M$. Let $\tilde{P}_w := \tilde{A}^{(w_L)} \cdots \tilde{A}^{(w_1)}$. Then*

$$\|\tilde{P}_w - P_w\|_{\infty,\mathrm{ent}} \le L\,(M + \varepsilon_A)^{L-1}\,\varepsilon_A,$$

and therefore the perturbed Doeblin margin satisfies

$$\delta'(\varepsilon_A) := \min_{i,j}(\tilde{P}_w)_{ij} \ge \delta - L\,(M + \varepsilon_A)^{L-1}\,\varepsilon_A.$$

In particular, any ε_A such that $\delta'(\varepsilon_A) > 0$ guarantees that the same *word w remains a valid Doeblin witness in the perturbed instance, with margin at least $\delta'(\varepsilon_A)$.*

Remark 25.9. *For relative entrywise perturbations, one may rewrite Lemma 25.8 with ε_A interpreted as a relative factor and with M replaced by the corresponding upper envelope on the block. The robust certificate must record which perturbation metric is used and the bound actually fed into the estimate.*

25.4 Robust certificate object

We now freeze the schema.

Definition 25.10 (Robust certificate object). *A* robust certificate *is a structured object containing:*

- *a model fingerprint (`model_hash`) and the perturbation metric (`perturbation_metric`);*
- *the calibration data (u, λ_*) used for reduced costs, together with the tight set H;*
- *margins and thresholds:*
  - *`tight_gap_gamma` storing γ from Definition 25.5;*

– ***eps_tight*** *such that* $\varepsilon_c <$ ***eps_tight*** *implies Lemma 25.6;*

– ***support_stability.eta_pos_min*** *storing* η *from Definition 25.3 and* ***support_stability.eps_support*** *storing* $\varepsilon_{\mathrm{supp}}$*;*

– ***doeblin.word*** *storing* w*,* ***doeblin.L*** *storing* L*, and* ***doeblin.delta*** *storing* δ*;*

– ***doeblin.M*** *storing the bound* M *from Definition 25.7;*

– ***eps_doeblin*** *such that* $\varepsilon_A <$ ***eps_doeblin*** *implies* $\delta'(\varepsilon_A) > 0$ *in Lemma 25.8;*

– *a rate pack section that records the contraction constants used in the unperturbed instance and provides explicit degraded constants as functions of* $(\varepsilon_c, \varepsilon_A)$ *whenever* $\varepsilon \le \varepsilon^\star$*.*

- *the aggregate robust radius* ***eps_star****: a pair* $\varepsilon^\star = (\varepsilon_c^\star, \varepsilon_A^\star)$ *satisfying*

$$\varepsilon_c^\star \le \texttt{eps\_tight}, \qquad \varepsilon_A^\star \le \texttt{eps\_doeblin}, \qquad \varepsilon_A^\star \le \texttt{support\_stability.eps\_support},$$

and any additional thresholds required by later robustness layers (Doeblin Stability and Recurrence Stability).

25.5 Concrete JSON schema (pipeline-facing)

The TeX contract above is mirrored by a pipeline-facing JSON schema and an example instance.

- `robust_certificate_schema.json`
- `robust_certificate_example.json`

For readability, we also display a concise schema skeleton here:

```
{
  "model_hash": "string",
  "perturbation_metric": {
    "cost_norm": "linfty",
    "matrix_metric": "absolute_entrywise|relative_entrywise",
    "matrix_norm": "linfty_entrywise",
    "support_preserving": "boolean"
  },
  "calibration": {
    "lambda_star": "number",
    "u_potential": "array[number]"
  },
  "support_stability": {
    "support_model": "support_preserving",
    "eta_pos_min": "number",
    "eps_support": "number"
  },
  "tight_core": {
    "tight_edges": "array[edge_id]",
    "tight_gap_gamma": "number",
    "tight_lipschitz_Kc": "number",
    "eps_tight": "number",
```

```
    "tight_graph_hash": "string_or_null"
  },
  "doeblin": {
    "word": "array[template_id]",
    "L": "integer",
    "M_bound": "number",
    "U_bound": "number",
    "delta": "number",
    "eps_doeblin": "number",
    "delta_prime_formula": "string",
    "U_prime_formula": "string"
  },
  "recurrence": {
    "mode": "combinatorial|hypothesis",
    "eps_recurrence": "number",
    "details": "object"
  },
  "rate_pack": {
    "projective_diameter_bound": "number",
    "tau_unperturbed": "number",
    "tau_prime_formula": "string",
    "eps_rate": "number",
    "kappa_prime_formula": "string"
  },
  "eps_star": {
    "eps_c_star": "number",
    "eps_A_star": "number"
  }
}
```

Remark 25.11. *Every later statement of the form "the conclusion survives perturbation" must cite: a margin field, the chosen perturbation metric, an explicit computed threshold, and the degraded constants produced when $\varepsilon \le \varepsilon^\star$. No other robustness language is permitted.*

26 Tight-Core Stability Theorem

This section upgrades the openness lemma of Robust Certificate Contract into a purely combinatorial invariance theorem: under an explicit margin condition, the *tight edge set*, its SCC decomposition, and its periodicity labels are unchanged throughout a perturbation neighborhood.

26.1 Stability statement for the tight-edge sign test

Fix a calibrated pair (u, λ_*) and reduced costs c_u as in (2). Let $H := \{e \in E : c_u(e) = 0\}$ be the tight edge set and let

$$\gamma := \min\{c_u(e) : e \in E \setminus H\} \in [0, +\infty)$$

(with the convention that $\gamma := 0$ when $H = E$).

Lemma 26.1 (Tight-set invariance under cost perturbations). *Let $\tilde{c} = c + \Delta c$ with $\|\Delta c\|_\infty \le \varepsilon_c$ and keep (u, λ_*) fixed. If $\varepsilon_c < \gamma/2$, then*

$$\tilde{H} := \{e \in E : \tilde{c}_u(e) \le \gamma/2\} = H,$$

and the inequalities

$$e \in H \implies \tilde{c}_u(e) \le \gamma/2, \qquad e \notin H \implies \tilde{c}_u(e) \ge \gamma/2$$

hold.

Proof. This is exactly the sign-stability conclusion of Lemma 25.6, with the explicit convention that $\gamma = 0$ forces the stable radius to be 0 in the degenerate all-tight case. □

26.2 Combinatorial invariance: SCC structure and periodicity labels

Let $G^{(0)} = (V, E^{(0)})$ be the directed graph obtained by restricting $G = (V, E)$ to the tight edges $E^{(0)} := H$. Write $\mathrm{SCC}(G^{(0)})$ for its SCC decomposition. For an SCC C we write $\mathrm{per}(C)$ for its period (the gcd of lengths of directed cycles in C).

Theorem 26.2 (Tight-core structural stability). *Assume $\gamma > 0$ and let $\tilde{c} = c + \Delta c$ with $\|\Delta c\|_\infty \le \varepsilon_c < \gamma/2$. Define the perturbed tight set by the stable sign test*

$$\tilde{H} := \{e \in E : \tilde{c}_u(e) \le \gamma/2\}.$$

Then:

- $\tilde{H} = H$ *(tight edge set is unchanged),*
- *the SCC decomposition of the tight graph is unchanged:* $\mathrm{SCC}(\tilde{G}^{(0)}) = \mathrm{SCC}(G^{(0)})$,
- *every SCC period is unchanged: for each tight SCC C,*

$$\mathrm{per}_{\tilde{G}^{(0)}}(C) = \mathrm{per}_{G^{(0)}}(C),$$

  so the periodic/aperiodic labels of SCCs are invariant.

Proof. The first bullet is Lemma 26.1. Once the edge set of the tight subgraph is identical, the directed graph itself is identical, so the SCC decomposition and the cycle-length sets within each SCC are identical. Hence the gcd defining the period is unchanged. □

Remark 26.3 (What this theorem does and does not claim). *Theorem 26.2 is a* certificate-level *stability statement. It asserts that all downstream logic that depends only on the tight graph (SCCs, periods, face-map checks, Doeblin search, avoidance graphs, and the OG1–OG4 branching) is robust throughout the neighborhood $\|\Delta c\|_\infty < \gamma/2$,* provided the certificate keeps (u, λ_*) fixed.

It does not *assert that the globally optimal calibrated pair for the perturbed instance is the same, nor that the minimizing set for the perturbed instance coincides with the unperturbed one. Those are separate questions; the robustness contract is that the* original witness object *remains valid and produces valid conclusions for all perturbations within the certified radius.*

26.3 Pipeline-facing fields added by Tight-Core Stability

Tight-Core Stability adds the following required fields to the `tight_core` section of the robust certificate:

- `tight_lipschitz_Kc`: a constant $K_c \geq 1$ such that $|\tilde{c}_u(e) - c_u(e)| \leq K_c \|\Delta c\|_\infty$ for all edges when (u, λ_*) is kept fixed (in the basic cost-perturbation model, $K_c = 1$),
- `eps_tight`: the computed stability threshold

$$\texttt{eps\_tight} = \frac{\gamma}{2K_c},$$

  so that $\|\Delta c\|_\infty < \texttt{eps\_tight}$ implies tight-core invariance,
- an optional `tight_graph_hash` (a fingerprint of the tight edge set) and `tight_scc_summary` (counts and period labels) for auditability.

These fields enforce the hard acceptance gate:

Every later robustness claim about the tight core must cite `tight_gap_gamma`, `tight_lipschitz_Kc`, and `eps_tight`.

27 Doeblin Stability and Robust Rate Pack

Robust Certificate Contract records a Doeblin word witness $w = w_1 \cdots w_L$ and its unperturbed margin $\delta = \min_{i,j}(P_w)_{ij} > 0$ on the relevant core/face block, where $P_w = A^{(w_L)} \cdots A^{(w_1)}$. Doeblin Stability upgrades this to a *robust rate pack*: the *same witness word* persists under small perturbations, and the associated Hilbert/Birkhoff contraction constants degrade by an explicit computable formula.

Throughout this section we work under the perturbation metric fixed in the robust certificate. We write ε_A for the template perturbation radius and assume the tight core is held fixed throughout the neighborhood by Tight-Core Stability (so the witness word remains admissible in the same tight SCC).

27.1 Product perturbation bounds and a surviving Doeblin margin

Assume the certificate stores a uniform entry bound $M \geq 1$ such that $\max_{k,i,j} A_{ij}^{(k)} \leq M$ on the block where the Doeblin witness is evaluated.

Lemma 27.1 (Entrywise product perturbation bound)**.** *Assume* absolute entrywise *perturbations* $\|\Delta A\|_{\infty,\mathrm{ent}} \leq \varepsilon_A$. *Let* $\tilde{P}_w := \tilde{A}^{(w_L)} \cdots \tilde{A}^{(w_1)}$. *Then*

$$\|\tilde{P}_w - P_w\|_{\infty,\mathrm{ent}} \leq L\,(M + \varepsilon_A)^{L-1}\,\varepsilon_A.$$

Consequently, the perturbed Doeblin margin satisfies

$$\delta'(\varepsilon_A) := \min_{i,j}(\tilde{P}_w)_{ij} \geq \delta - L\,(M + \varepsilon_A)^{L-1}\,\varepsilon_A.$$

Proof. This is Lemma 25.8, restated for later reuse. □

Remark 27.2 (Relative entrywise perturbations)**.** *For* relative entrywise *perturbations* $|\Delta A_{ij}^{(k)}| \leq \varepsilon_A A_{ij}^{(k)}$*, one may convert to an absolute bound on the relevant block:* $\|\Delta A\|_{\infty,\mathrm{ent}} \leq \varepsilon_A M$*. All formulas below remain valid after replacing* ε_A *by* $\varepsilon_A M$ *in the right-hand sides and recording this choice in the certificate.*

27.2 A robust diameter bound and Birkhoff contraction coefficient

In addition to δ, the robust certificate stores an *upper* bound $U := \max_{i,j}(P_w)_{ij}$ on the same block (or any certified upper envelope used by the pipeline).

Lemma 27.3 (Upper envelope under perturbations)**.** *Under the assumptions of Lemma 27.1,*

$$U'(\varepsilon_A) := \max_{i,j}(\tilde{P}_w)_{ij} \leq U + L\,(M + \varepsilon_A)^{L-1}\,\varepsilon_A.$$

Proof. Entrywise, $|(\tilde{P}_w)_{ij} - (P_w)_{ij}| \leq \|\tilde{P}_w - P_w\|_{\infty,\mathrm{ent}}$. Taking maxima and applying Lemma 27.1 gives the claim. □

Lemma 27.4 (Robust projective diameter bound for a positive block)**.** *Assume $\delta'(\varepsilon_A) > 0$ and let $U'(\varepsilon_A)$ be as in Lemma 27.3. Then the Hilbert projective diameter of the image of the positive cone under $\tilde{P}_w$ satisfies*

$$\Delta(\varepsilon_A) \;\leq\; 2\,\log\Big(\frac{U'(\varepsilon_A)}{\delta'(\varepsilon_A)}\Big).$$

In particular, the Birkhoff contraction coefficient of $\tilde{P}_w$ satisfies

$$\tau'(\varepsilon_A) \;\leq\; \tanh\Big(\frac{\Delta(\varepsilon_A)}{4}\Big).$$

Proof. If all entries of a positive matrix lie in $[\delta', U']$, then $\max_{i,j,k,\ell} \frac{B_{ik}B_{j\ell}}{B_{i\ell}B_{jk}} \leq (U'/\delta')^2$. The standard diameter bound is $\Delta(B) \leq \log\big(\max \frac{B_{ik}B_{j\ell}}{B_{i\ell}B_{jk}}\big)$, giving $\Delta \leq 2\log(U'/\delta')$. The Birkhoff bound then yields $\tau \leq \tanh(\Delta/4)$. □

27.3 Robust rate extraction under a recurrence certificate

Doeblin Stability does *not* assert recurrence. Instead, it states: once recurrence is certified or assumed as a named hypothesis, the contraction rate survives perturbation with explicit degradation.

Proposition 27.5 (Robust contraction per Doeblin occurrence)**.** *Assume $\delta'(\varepsilon_A) > 0$. Whenever the Doeblin word w occurs (so the block product applied at that time is $\tilde{P}_w$), the Hilbert distance contracts by at least the factor $\tau'(\varepsilon_A)$ from Lemma 27.4.*

Proof. This is the Birkhoff contraction principle applied to the positive block $\tilde{P}_w$. □

Corollary 27.6 (Rate degradation under syndetic recurrence)**.** *Assume a recurrence certificate provides a bounded-gap modulus B for occurrences of w along the trajectory under study. Let $L = |w|$ and suppose $\delta'(\varepsilon_A) > 0$. Then for all times n,*

$$d_H(x_n, y_n) \leq C\,(\tau'(\varepsilon_A))^{\lfloor n/(L+B)\rfloor}\,d_H(x_0, y_0),$$

so an admissible exponential rate is

$$\kappa'(\varepsilon_A) = -\frac{\log \tau'(\varepsilon_A)}{L+B}.$$

Proof. Partition the trajectory into consecutive windows of length at most $L+B$, each containing an occurrence of w. Each window contributes at least one contraction by Proposition 27.5; absorbing the inter-block projective nonexpansive bounds into a prefactor yields the stated inequality. □

27.4 Pipeline-facing fields added by Doeblin Stability

Doeblin Stability requires the robust certificate to store enough information to compute:

- the surviving Doeblin margin function $\delta'(\varepsilon_A)$ and a computed threshold `eps_doeblin` ensuring $\delta'(\varepsilon_A) > 0$,
- an upper envelope U for the unperturbed Doeblin block and a formula for $U'(\varepsilon_A)$,
- a diameter bound $\Delta(\varepsilon_A)$ and a contraction factor bound $\tau'(\varepsilon_A)$,
- an optional `eps_rate` threshold under which a target contraction quality is guaranteed (e.g., $\delta'(\varepsilon_A) \geq \delta/2$ and $U'(\varepsilon_A) \leq 2U$).

These enforce the hard acceptance gate:

No statement may claim robust survival of a Doeblin witness or a rate without citing `doeblin.delta`, `doeblin.U_bound`, and the computed thresholds `eps_doeblin` and (when used) `eps_rate`.

28 Recurrence Robustness Layer (Robust Mode)

The gap-free upgrade separates *existence* of an algebraically contracting block (a Doeblin word on the active face) from *recurrence* of that block along trajectories. A correct finite recurrence layer is obtained by separating two objects: **(RC1)** is a purely combinatorial predicate for tight trajectories in a fixed tight SCC H, while $\mathrm{OG4}(H, w)$ records the explicit failure mode via an avoidance-cycle witness. For maximal determinism, the pipeline may optionally replace the single word w by the *canonical Doeblin family* $\mathcal{B}(H)$ and use the multi-block predicate **(MB–RC1)** (Section 13); the robustness logic is identical because the avoidance graphs are still finite combinatorial objects.

This section enforces that separation in a robust, checkable way. All recurrence statements are permitted only in one of two modes, each with an explicit robustness threshold.

28.1 Two recurrence modes with margins

Fix an aperiodic tight SCC H of the tight graph $G^{(0)}$ and a certified Doeblin word w supported in H.

Definition 28.1 (Recurrence modes for a robust certificate)**.** *A* robust certificate *records recurrence information in exactly one of the following modes.*

- ***Mode (combinatorial).*** *The certificate records that **(RC1)** holds for (H, w) (Definition 14.1), with an explicit syndetic modulus $L_{\mathrm{syn}}(H, w)$ (Lemma 14.6). Equivalently, the avoidance graph $\mathcal{A}_{\neg w}(H)$ is acyclic.*
- ***Mode (hypothesis).*** *The certificate records a user-supplied recurrence hypothesis on a* chosen *minimizing trajectory, for example a bounded-gap hypothesis (gap bound B) or a lower-density hypothesis (density margin $\eta > 0$). In this mode, the pipeline does* not *certify the truth of the hypothesis; it only certifies that all constants depending on w and the template family degrade explicitly and continuously under perturbation.*

28.2 Robustness threshold for recurrence statements

The point of robust mode is that the same finite witness objects remain valid on an explicit neighborhood. The neighborhood size is computed from earlier margins.

Definition 28.2 (Recurrence robustness radius)**.** *Let $\varepsilon_{\text{tight}}$ be the tight-core stability threshold from Tight-Core Stability (Theorem in Section 26) and let $\varepsilon_{\text{Doeblin}}$ be the Doeblin-survival threshold from Doeblin Stability (Section 27). Define*

$$\varepsilon_{\text{rec}} := \min\{\varepsilon_{\text{tight}}, \varepsilon_{\text{Doeblin}}\}.$$

A recurrence statement is said to be robustly certified *on the neighborhood of size ε_{rec} if it is in Mode (combinatorial) and its finite witness objects are unchanged throughout that neighborhood.*

Proposition 28.3 (Stability of the RC1/OG4 dichotomy)**.** *Assume $\varepsilon \leq \varepsilon_{\text{tight}}$. Then the tight graph $G^{(0)}$ and its SCC decomposition are unchanged under the perturbation model, so the avoidance graph $\mathcal{A}_{\neg w}(H)$ is unchanged as a finite directed graph. Consequently, the dichotomy*

$$\textit{(RC1) holds for } (H, w) \quad \textit{or} \quad \textit{OG4}(H, w) \textit{ holds}$$

(and any explicit witness: either L_{syn} or an avoidance cycle) is invariant under all such perturbations.

Proof. Under $\varepsilon \leq \varepsilon_{\text{tight}}$, Tight-Core Stability guarantees that the tight-edge set and hence the tight SCC H are unchanged. The avoidance graph $\mathcal{A}_{\neg w}(H)$ is constructed purely from the tight edge set of H and the fixed word w (Definition 14.2), so its vertex/edge set is unchanged. Therefore acyclicity (RC1) and the existence of an avoidance cycle (OG4) are unchanged, and any explicit witness carries over. □

Remark 28.4 (Why ε_{rec} uses $\varepsilon_{\text{Doeblin}}$)**.** *Proposition 28.3 is purely combinatorial and only needs $\varepsilon_{\text{tight}}$. However, in the contraction pipeline, recurrence is meaningful only in tandem with the Doeblin minorization property of w. Thus robust certificates record $\varepsilon_{\text{rec}} = \min\{\varepsilon_{\text{tight}}, \varepsilon_{\text{Doeblin}}\}$ so that "recurrence + Doeblin" survives together.*

28.3 What robust mode does and does not claim

- In Mode (combinatorial), recurrence is *derived* for *all* tight trajectories supported in the same SCC H, with an explicit gap bound L_{syn}. This derived recurrence is stable for all perturbations with $\varepsilon \leq \varepsilon_{\text{rec}}$.
- In Mode (hypothesis), recurrence is *not* asserted without the stated hypothesis. Robustness refers only to the *constants* in the contraction estimate (diameter bounds, Birkhoff factor, and rate degradation) when the same hypothesis is assumed along the perturbed instance.

Enforcement rule. No theorem, corollary, or remark may state a recurrence conclusion unless it cites *either* a Mode (combinatorial) certificate field (with L_{syn} and ε_{rec}) *or* a Mode (hypothesis) field (with its explicit margin and the disclaimer that the hypothesis is assumed).

29 Robustness of obstruction outcomes

The robustness layers Tight-Core Stability (tight-core stability) and Section 25.2 (support-pattern stability) also control the *NO* branches OG1–OG4. This section makes that explicit: when certification fails, the obstruction witness can still export a certified local invariance radius on which the obstruction label is provably stable.

29.1 Certified obstruction invariance radii

Fix a certified instance and the perturbation model stored in the pipeline artifacts. Write $\varepsilon_{\text{tight}} :=$ `tight_core.eps_tight` for the tight-core stability threshold and $\varepsilon_{\text{supp}} :=$ `support_stability.eps_support` for the support-pattern stability threshold.

Theorem 29.1 (Robustness of OG outcomes under support-preserving perturbations)**.** *Assume support-preserving perturbations (Definition 25.1). Let $\varepsilon = (\varepsilon_c, \varepsilon_A)$ satisfy*

$$\varepsilon_c < \varepsilon_{\text{tight}} \qquad \textit{and} \qquad \varepsilon_A < \varepsilon_{\text{supp}}.$$

Then each obstruction predicate OG1–OG4 is locally invariant on the ε-neighborhood, in the following sense.

- *If OG1 holds for the unperturbed instance, then OG1 holds for every perturbed instance in the neighborhood. Moreover, the tight SCC period table provided by the OG1 witness remains valid throughout the neighborhood (the tight SCC decomposition and periods do not change under $\varepsilon_c < \varepsilon_{\text{tight}}$).*
- *If OG2 holds for the unperturbed instance with witness cycle data $(H, cycle\_word, i_{\text{out}}, j_{\text{in}})$, then the same witness remains valid throughout the neighborhood. Tight-core stability keeps the same SCC H, and support-pattern stability preserves the template sparsity patterns used in the face-map check, so the inconsistency cycle remains an inconsistency cycle.*
- *If OG3 holds for the unperturbed instance with either a cut witness or an automaton non-reachability witness, then OG3 holds throughout the neighborhood and the same witness remains valid. Indeed, Doeblin-word existence on $S(H)$ is a reachability predicate in the Boolean-pattern product automaton built from support patterns, so support-pattern invariance keeps this predicate unchanged.*
- *If OG4 holds for the unperturbed instance with a directed avoidance cycle in $\mathcal{A}_{\neg w}(H)$, then OG4 holds throughout the neighborhood and the same avoidance cycle remains a valid witness. The avoidance graph is built from the same tight SCC and the same support patterns; both are invariant under the stated bounds.*

Consequently, in every NO branch OG1–OG4 the pipeline may attach a certified local invariance radius

$$\varepsilon^{\star}_{\text{NO}} := \big(\varepsilon_{\text{tight}},\, \varepsilon_{\text{supp}}\big),$$

recorded in the obstruction witness `local_invariance` *field.*

Proof. Under $\varepsilon_c < \varepsilon_{\text{tight}}$, Tight-Core Stability keeps the tight edge set, tight graph, SCC decomposition, and SCC periods unchanged. Under $\varepsilon_A < \varepsilon_{\text{supp}}$ and support-preserving perturbations, Lemma 25.4 keeps the Boolean support patterns unchanged on the relevant blocks. Each OG predicate is then a purely combinatorial predicate on these finite structures: OG1 is the absence of period-1 SCCs, OG2 is a finite inconsistency certificate for the face map, OG3 is non-reachability in a fixed finite automaton, and OG4 is the existence of a directed cycle in a fixed finite avoidance graph. Therefore the truth of each predicate and the validity of each stored witness are invariant on the neighborhood. □

Remark 29.2 (When the NO-radius is not certified). *If the perturbation model allows support creation on previously zero entries (i.e., perturbations are not support-preserving), then OG2/OG3/OG4 need not be locally invariant for any positive matrix radius, because new positive entries can create new Doeblin words or remove face inconsistencies. In that case, the obstruction witness should set* `local_invariance.certified=false` *and record only the cost-side radius (when available).*

30 Approximate Certification on Parameter Regions (Robust Neighborhood Cover)

This section upgrades pointwise certificates to *region certification*. The guiding observation is topological and quantitative: all certificate predicates used by the pipeline are *open* conditions once the robust margins are recorded. Therefore a finite verification on an ε-net certifies an entire parameter region except near an explicit exceptional set where margins vanish.

30.1 Local openness packaged as a single statement

Fix a parameter instance θ (costs and templates) and assume the pipeline produces a robust certificate object with robustness radii $\varepsilon_{\text{tight}}, \varepsilon_{\text{Doeblin}}, \varepsilon_{\text{rec}}$ and $\varepsilon_* = \min\{\varepsilon_{\text{tight}}, \varepsilon_{\text{Doeblin}}, \varepsilon_{\text{rate}}, \varepsilon_{\text{rec}}\}$ as in Sections 25–28.

Theorem 30.1 (Robust neighborhood invariance). *Let θ admit a robust certificate with radius $\varepsilon_* > 0$. Then every perturbed instance θ' with perturbation size $\leq \varepsilon_*$ satisfies:*

- *the tight graph $G^{(0)}$ and its SCC/period decomposition are unchanged;*
- *the stored Doeblin word w remains a valid Doeblin witness on the same active face, with explicit margin $\delta'(\varepsilon_*) > 0$;*
- *the recurrence layer is unchanged in Mode (combinatorial) (same RC1/OG4 outcome and witnesses), or remains a stated hypothesis in Mode (hypothesis);*
- *the projective contraction constants degrade explicitly according to the formulas stored in the certificate.*

In particular, every theorem in this paper that is invoked through certificate fields at θ holds verbatim at θ' with constants replaced by the certificate's degraded constants.

Proof. The tight-core invariance is Tight-Core Stability. Doeblin survival and the explicit $\delta'(\varepsilon)$ / $\tau'(\varepsilon)$ bounds are Doeblin Stability. Recurrence-layer stability in Mode (combinatorial) is Recurrence Stability. All remaining constants are explicit functions of these margins and the FTG bounds, hence are stable under the same inequalities. □

30.2 Finite net certification of a compact region

Let (Ω, d) be a compact parameter set with a product metric compatible with the perturbation model (e.g. $d(\theta, \theta') = \max\{\|\Delta c\|_\infty, \|\Delta A\|_{\infty,\text{entry}}\}$ in absolute-entrywise mode).

Definition 30.2 (ε-net). *A finite set $\mathcal{N} \subset \Omega$ is an ε-net if every $\theta \in \Omega$ lies within distance ε of some $\theta_i \in \mathcal{N}$.*

Theorem 30.3 (Region certification by robust balls)**.** *Suppose that for each net point $\theta_i \in \mathcal{N}$ the pipeline produces a robust certificate with radius $\varepsilon_*(\theta_i) > 0$. Define the certified region*

$$\Omega_{\mathrm{cert}} := \bigcup_{\theta_i \in \mathcal{N}} B\big(\theta_i, \varepsilon_*(\theta_i)\big) \cap \Omega.$$

Then every instance $\theta \in \Omega_{\mathrm{cert}}$ inherits the same certified conclusions as its center θ_i (in the sense of Theorem 30.1). Moreover, if $\mathcal{N}$ is an ε-net and $\min_i \varepsilon_*(\theta_i) \geq \varepsilon$, then $\Omega_{\mathrm{cert}} = \Omega$.

Proof. If $\theta \in \Omega_{\mathrm{cert}}$ then $\theta \in B(\theta_i, \varepsilon_*(\theta_i))$ for some center θ_i. Apply Theorem 30.1 at θ_i. The final claim follows directly from the definition of an ε-net. □

30.3 Exceptional set description

The only ways robustness can fail are the explicit margin-boundary events where $\varepsilon_* = 0$. Accordingly, region mode reports uncovered points as being near one of the following certificate-boundary loci:

- **Tightness degeneracy:** $\gamma = 0$ (tight/non-tight separation vanishes), so the tight graph can change.
- **Doeblin degeneracy:** $\delta = 0$ for the stored word (or no Doeblin word exists), so minorization can fail.
- **Recurrence boundary:** the RC1/OG4 dichotomy changes only when the tight graph changes (hence is subsumed by $\gamma = 0$), while Mode (hypothesis) has no derived recurrence boundary because recurrence is not asserted.

30.4 Pipeline: region mode outputs

In region mode, the pipeline:

- samples or constructs a finite net $\mathcal{N}$ of parameter points;
- runs the pointwise certificate pipeline at each θ_i;
- outputs the list of certified balls $\big(\theta_i, \varepsilon_*(\theta_i)\big)$ and their conclusions;
- reports uncovered points and diagnoses them in terms of the margin boundaries above.

A recommended machine-readable artifact is a `robust_region_report.json` containing: centers with their radii ε_*, the induced outcome labels (OG1–OG4 or contraction), and an uncovered-set diagnostic summary.

30.5 Finite frontier atlas extraction (chamberwise output)

Because every certificate predicate is open once margins are recorded (Theorem 30.1), each certified ball $B(\theta_i, \varepsilon_*(\theta_i))$ carries a locally constant *outcome label* in

$$\{\texttt{contraction}, \texttt{OG1}, \texttt{OG2}, \texttt{OG3}, \texttt{OG4}\}.$$

This permits an explicit finite atlas construction on Ω.

Definition 30.4 (Ball adjacency and chambers). *Given a region report with certified balls* $\{(\theta_i, \varepsilon_i, \ell_i)\}$*, define an undirected adjacency graph on indices by* $i \sim j$ *if the balls overlap:*

$$d(\theta_i, \theta_j) \le \varepsilon_i + \varepsilon_j.$$

A chamber *is a connected component of the induced subgraph on indices with the same label* ℓ_i*.*

Theorem 30.5 (Finite frontier atlas theorem). *Let* Ω *be compact and let region mode output certified balls with labels. Then:*

- *on every ball* $B(\theta_i, \varepsilon_i)$*, the outcome label is constant and the corresponding certificate conclusions hold throughout the ball (Theorem 30.1);*
- *merging overlapping balls by Definition 30.4 yields a finite chamber list, each chamber carrying a single verified outcome label and (when present) a witness payload that may be shared across the chamber;*
- *any uncovered portion of* Ω *is confined to the explicit boundary loci where margins collapse (notably* $\gamma \to 0$ *or* $\delta \to 0$*), and the report can summarize this via counted diagnostics.*

Accordingly, region mode produces an explicit cover-or-atlas *artifact: either a finite cover by contraction-labeled balls (yielding global contraction universality), or a boundary-atlas report that localizes precisely where and why uniform certification fails.*

Remark 30.6 (Maximal frontier report in region mode). *A region-mode atlas is "maximal" in the practical sense that every certified ball is labeled by the minimal failing gate at its center, and the uncovered diagnostics record which margin-collapse loci are responsible for missing coverage. This turns parameter exploration into a finite decision object: a chamber list plus a boundary summary, rather than an informal plot.*

31 From regional robustness to cover/atlas universality on compact normalized families

The robustness layer establishes a local principle: whenever a robust certificate exists at a parameter value θ, the same conclusions hold on an explicit neighborhood around θ, with conservatively degraded constants. This section records what is required to upgrade from "open neighborhoods away from degeneracy" to a global statement on a compact normalized parameter family.
Terminology. Throughout this section, "global" means *on a fixed compact normalized family* Θ, and "universality" is always understood *within the certified class* in the *cover-or-atlas* sense: either good-global contraction is certified, or an explicit boundary-atlas report localizes the obstruction/degeneracy set.

31.1 Two global targets

There are two distinct global goals.

- **Global regime decidability (already achieved on the certified class).** For any instance in the certified class, the pipeline halts and returns a finite outcome object: *either* a robust contraction certificate, *or* a concrete obstruction witness (OG1–OG4).

- **Good-global contraction universality (stronger).** On a parameter region Θ, every parameter point lies in the contraction regime, so the exceptional set is empty and one can state a single uniform robustness radius on all of Θ.

The remainder of this section concerns the second target.

31.2 Paired radii convention

In the robust certificate contract, perturbations may act on two layers: costs and template entries. Accordingly, the local robustness radius naturally comes as a pair

$$\boldsymbol{\varepsilon}^*(\theta) := (\varepsilon_c^*(\theta),\, \varepsilon_A^*(\theta)),$$

meaning that the certificate remains valid whenever both perturbations satisfy $\|\Delta c\| \le \varepsilon_c^*(\theta)$ and $\|\Delta A\| \le \varepsilon_A^*(\theta)$ in the Robust Certificate Contract metric. For compactness and cover arguments, it is often convenient to also record the conservative scalar floor

$$\varepsilon_{\min}^*(\theta) := \min\{\varepsilon_c^*(\theta),\, \varepsilon_A^*(\theta)\}.$$

This scalar floor is sufficient for a single-radius cover statement, while the pair is the natural artifact for downstream reproducibility.

31.3 Uniform margins and a global contraction theorem

Let Θ be a compact parameter region. For each $\theta \in \Theta$, the reduction stage produces a tight graph $H(\theta)$, face data, and (when obstruction gates do not fire) a Doeblin witness word $w(\theta)$ on an aperiodic tight SCC.

Definition 31.1 (Uniform robust margins on a region)**.** *We say that Θ admits* uniform robust margins *for a fixed witness word w if there exist constants $\gamma_0, \delta_0 > 0$ and a fixed aperiodic tight SCC H such that for every $\theta \in \Theta$:*

- *the tightness gap satisfies $\gamma(\theta) \ge \gamma_0$ and the tight-core graph equals $H(\theta) = H$;*
- *the Doeblin witness product along w satisfies $\min(P_w(\theta)) \ge \delta_0$ on the active coordinates;*
- *the recurrence gate holds on H: either RC1 holds for (H, w) in single-block mode, or MB–RC1 holds for H in canonical multi-block mode (equivalently: the appropriate avoidance automaton contains no directed cycle).*

Theorem 31.2 (Good-global contraction universality on Θ from uniform margins)**.** *Assume Θ is compact and normalized. If Θ admits uniform robust margins in the sense of Definition 31.1 for a fixed Doeblin witness object, then there exist:*

- *a paired uniform robustness radius $\boldsymbol{\varepsilon}_{\mathrm{global}} = (\varepsilon_{c,\mathrm{global}}, \varepsilon_{A,\mathrm{global}})$,*
- *a conservative scalar floor $\varepsilon_{\min,\mathrm{global}} := \min\{\varepsilon_{c,\mathrm{global}}, \varepsilon_{A,\mathrm{global}}\} > 0$,*
- *and a uniform rate pack $(\tau_{\mathrm{global}}, C_{\mathrm{global}}, \kappa_{\mathrm{global}})$,*

with the following property. For every $\theta \in \Theta$:

- *every tight trajectory in the aperiodic SCC H experiences Doeblin blocks with a bounded-gap modulus that is uniform over Θ;*

- *consequently, the associated projective dynamics contract uniformly with rate $\tau_{\mathrm{global}} < 1$ and prefactor C_{global};*

- *the same conclusions remain valid under all perturbations satisfying $\|\Delta c\| \le \varepsilon_{c,\mathrm{global}}$ and $\|\Delta A\| \le \varepsilon_{A,\mathrm{global}}$ (Robust Certificate Contract metric), and in particular for all perturbations of size at most $\varepsilon_{\min,\mathrm{global}}$ in either layer.*

Moreover, $\boldsymbol{\varepsilon}_{\mathrm{global}}$ is computable from the uniform margins (γ_0, δ_0) and the fixed witness metadata (word length and uniform entry bounds) by taking the componentwise minimum of the explicit thresholds from Tight-Core Stability and Doeblin Stability.

Proof. By Tight-Core Stability (tight-core stability), any perturbation smaller than the explicit threshold determined by γ_0 preserves the tightness classification and the SCC decomposition. By Doeblin Stability (Doeblin stability), the fixed Doeblin product along w with entry floor δ_0 persists under perturbations below the explicit telescoping-product threshold, yielding a degraded floor δ_0'. Since H and the chosen witness object are fixed throughout Θ, the appropriate avoidance automaton (single-block $\mathsf{A}(H, w)$ or multi-block $\mathsf{A}(H, \mathcal{B}(H))$) is fixed. RC1 holding implies that this finite automaton has no directed cycle; hence every tight trajectory must hit the Doeblin block within at most $L_{\mathrm{syn}}(H, w) \le |V(\mathsf{A}(H, w))|$ steps (Corollary 14.7), giving a uniform recurrence modulus. Combining the recurrence modulus with the Hilbert/Birkhoff contraction lemma and the explicit rate pack yields uniform projective contraction with constants depending only on δ_0' and the certificate geometry. Taking the componentwise minimum of the explicit Tight-Core Stability and Doeblin Stability thresholds produces a paired radius valid for all $\theta \in \Theta$. □

31.4 A global theorem phrased in terms of degeneracy avoidance

The uniform-margin hypothesis is clean, but one still wants a practical route to verify it. A useful middle step is to express good-global contraction universality on compact normalized Θ within the certified class as the absence of certain degeneracy surfaces.

Definition 31.3 (Degeneracy sets)**.** *Fix a witness word w and consider a region Θ on which the tight SCC H and the active coordinates are well-defined. Define the following subsets of Θ:*

- $\mathcal{D}_\gamma := \{\theta \in \Theta : \gamma(\theta) = 0\}$ *(tightness degeneracy / tie surfaces),*

- $\mathcal{D}_\delta(w) := \{\theta \in \Theta : \min(P_w(\theta)) = 0\}$ *(loss of strict positivity for the witness product),*

- $\mathcal{D}_{\mathrm{RG}} := \{\theta \in \Theta :$ *RC1 fails for* $(H, w)\}$ *(recurrence-forcing failure).*

Proposition 31.4 (Good-global contraction universality from empty degeneracy on Θ)**.** *Assume Θ is compact and that H and w are constant on Θ. If*

$$\mathcal{D}_\gamma = \emptyset, \qquad \mathcal{D}_\delta(w) = \emptyset, \qquad \mathcal{D}_{\mathrm{RG}} = \emptyset,$$

then Θ admits uniform robust margins and Theorem 31.2 applies. In particular, γ and $\theta \mapsto \min(P_w(\theta))$ have strictly positive minima on Θ.

Proof. On a region where H and w are fixed, $\gamma(\theta)$ is the minimum of finitely many strict reduced-cost separations among edges in the fixed reduced instance, hence is continuous. The map $\theta \mapsto \min(P_w(\theta))$ is continuous because $P_w(\theta)$ is a finite product of matrices whose entries depend continuously on θ. On a compact set, each continuous function attains its minimum. If the corresponding zero set is empty, the minimum must be strictly positive, giving constants $\gamma_0, \delta_0 > 0$. RC1 being satisfied everywhere on Θ provides the recurrence-forcing item in Definition 31.1. □

For the linear parameter families treated in §35, the parameter space decomposes into finitely many polyhedral chambers on which the tight graph and the witness metadata are constant. In that setting, Proposition 31.4 can be applied chamberwise, and good-global contraction universality on compact normalized Θ within the certified class reduces to taking minima over the finitely many certified chambers.

31.5 Region-mode coverage as a certificate of good-global contraction on compact normalized families

Cover/Atlas Principle provides a covering principle: each robust certificate at θ certifies a whole neighborhood around θ. A global statement on Θ follows once the union of certified neighborhoods covers Θ.

Proposition 31.5 (Good-global contraction certificate via region coverage). *Assume Θ is compact and normalized. Suppose region mode produces a finite family of robust certificates $\{\mathcal{C}_i\}$ with certified paired radii $\boldsymbol{\varepsilon}^*(\mathcal{C}_i)$ such that*

$$\Theta \subset \bigcup_i B(\theta_i, \varepsilon^*_{\min}(\mathcal{C}_i)),$$

where $B(\theta, \varepsilon)$ denotes the ball in parameter space used for sampling and $\varepsilon^*_{\min}$ is the conservative scalar floor. Then Θ admits a good-global contraction certificate *within the certified class on the compact normalized family Θ. One may set*

$$\gamma_0 := \min_i \gamma(\mathcal{C}_i), \qquad \delta_0 := \min_i \delta(\mathcal{C}_i), \qquad \boldsymbol{\varepsilon}_{\text{global}} := \big(\min_i \varepsilon^*_c(\mathcal{C}_i),\ \min_i \varepsilon^*_A(\mathcal{C}_i)\big),$$

and conclude uniform contraction on all of Θ with explicit degraded constants.

Proof. The conclusion is exactly the Cover/Atlas Principle covering theorem applied to the finite subcover exhibited by region mode. Uniformity follows by taking minima over the finite family. □

Proposition 31.6 (Finite termination of region mode under a uniform scalar floor). *Assume $\Theta \subset \mathbb{R}^d$ is compact and normalized and admits a uniform lower bound*

$$\inf_{\theta \in \Theta} \varepsilon^*_{\min}(\theta) \geq \varepsilon_0 > 0$$

for the conservative scalar floors returned by the pipeline. Then a grid net of mesh $\varepsilon_0/2$ produces a finite cover, hence region mode terminates with a good-global contraction certificate within the certified class on the compact normalized family Θ.

Proof. Compactness implies Θ can be covered by finitely many balls of radius $\varepsilon_0/2$. At each ball center, the pipeline returns a certificate with scalar floor at least ε_0. Therefore each such certificate neighborhood contains the corresponding grid ball, giving a finite cover of Θ. □

31.6 A practical push: hybrid chamberwise verification

The most practical route to good-global contraction universality on compact normalized families (within the certified class) is a hybrid procedure:

- use the chamber mechanism to identify regions on which (H, w) are constant,
- on each chamber, either prove degeneracy sets are empty (Proposition 31.4), or close coverage via region mode (Proposition 31.5),

- if coverage fails, the uncovered pieces are automatically labeled as "near degeneracy" and come with obstruction witnesses.

The key point is that the framework never hides the boundary: good-global contraction universality on compact normalized Θ within the certified class is exactly the claim that all named degeneracies are absent on Θ. When they are present, the pipeline returns a concrete finite witness locating them.

Algorithm 9 Hybrid cover/atlas check on compact normalized Θ within the certified class (certificate-driven)

Require: Compact region Θ (parameterization), perturbation metric (Robust Certificate Contract contract)
Ensure: Either a GlobalUniversalityReport for compact normalized Θ within the certified class, or an explicit near-degeneracy/obstruction report

```
Partition Θ into chambers where (H, w) are constant (by the tight-core/chamber mechanism)
for each chamber Ω do
    Run region mode on Ω with adaptive refinement
    if coverage closes then
        Record chamberwise global bounds γ_0(Ω), δ_0(Ω), ε_global(Ω)
    else
        Record uncovered set and its near-degeneracy label (tie surface, positivity collapse, or recurrence-gate failure)
    end if
end for
If all chambers close coverage, take minima of chamberwise bounds to form a global report on Θ
```

Remark 31.7 (What is and is not proved "globally"). *The results above provide two globally valid statements.*

- *On the certified class, the pipeline provides a global decidability frontier: it always returns a finite obstruction or a finite robust contraction certificate.*
- *On a parameter family Θ, a good-global contraction universality statement is achieved when either uniform margins are proved or region-mode coverage closes.*

If neither occurs, the output is still constructive: one gets an explicit description of where and how universality can fail.

31.7 Frontier atlas compression: a single headline artifact for region mode

Region mode produces a full audit object (the global report) that is intentionally verbose. For operational use one usually wants a single object that answers the frontier questions directly: which regimes occur, which minimal gates are responsible, and where the boundary loci concentrate.

Proposition 31.8 (Canonical compression map: global report $\to$ frontier atlas report). *There exists a deterministic compression map*

$$\mathsf{CompressToFrontierAtlas} : \mathtt{GlobalUniversalityReport} \longrightarrow \mathtt{FrontierAtlasReport}$$

with the following properties.

- *The output* `FrontierAtlasReport` *records a chamber list labeled by* `contraction`*, OG1–OG4, and* `near_degeneracy`*, together with explicit references to the supporting certificates and witnesses.*

- *Each chamber also exports a single* `chamber_invariance.eps_dec` *value: a certified decision-radius lower bound computed as the minimum local invariance radius among the chamber's certified subcover items (*`chamber_invariance.cover_refs`*). This makes the atlas itself a carry-forward certificate object for region-mode workflows.*

- *The output includes a* `coverage` *block stating whether the certified subcover closes on the whole region and, if not, localizing the uncovered remainder inside named degeneracy loci, as well as a* `stratification` *block grouping chamber IDs by outcome label.*

- *The output records the boundary loci implicated by the report (in particular: $\gamma \to 0$, $\delta \to 0$, and recurrence-gate failure when applicable) and includes localization notes supplied by region mode.*

- *If the global report certifies good-global contraction universality (empty uncovered set), then the frontier atlas report contains only* `contraction` *labels and can be treated as trivial.*

- *If the global report is a boundary-global atlas, then every noncontraction label in the frontier atlas report is supported by either an explicit obstruction witness (OG1–OG4) or by a declared degeneracy neighborhood in which the relevant margin collapses.*

Proof. The global report already contains exactly the required ingredients: certificate IDs for covered balls, obstruction witness references when obstruction gates fire, and a near-degeneracy diagnostic block when coverage cannot close outside declared degeneracy neighborhoods. The compression map simply: (i) discards net bookkeeping, (ii) merges overlapping certified balls into chamber labels when chamber identifiers are available, (iii) computes `chamber_invariance.eps_dec` as the minimum of the local invariance radii of the subcover items listed in `chamber_invariance.cover_refs`, (iv) normalizes the union of the near-degeneracy diagnostics into a list of loci, and (v) records an index of referenced certificate and witness files. No new mathematical claim is introduced by this compression. □

Remark 31.9 (OP–ATLAS–FIN as an output statement)**.** *Within the certified class on compact normalized families, Proposition 31.8 makes the "cover-or-atlas" principle a single exported artifact: the frontier atlas report. This is the preferred object for downstream work that needs a stable, witness-producing map of the decidable island boundary on a region.*

31.8 Beyond compact families: normalization for cover/atlas universality within the certified class

When one asks for cover/atlas universality beyond a fixed compact normalized family, the obstruction is usually not conceptual but topological: the raw parameter space is typically *noncompact* (because of scaling degrees of freedom, unbounded costs, or templates whose entries can blow up or collapse to zero). The cover/atlas conclusions above are therefore best viewed as *compact* statements within the certified class.

In many positive-template models, however, the noncompactness is largely artificial. Two standard reductions convert noncompact questions into compact ones:

- **Projectivization/scaling.** The extremal growth rate is homogeneous under simultaneous scaling of templates. Working in projective coordinates (or fixing a canonical scale) removes this degree of freedom.
- **Canonical normalization.** One may fix a normalization such as $\max_{i,j} A_k(i,j) = 1$ (entrywise), or $\|A_k\|_1 = 1$ (column-stochastic scaling), provided the admissible class is closed under the chosen normalization and the pipeline invariants are scale-invariant.

The practical upshot is that a cover/atlas universality theorem can often be stated as a compact cover/atlas universality theorem on a normalized parameter space within the certified class, together with an explicit description of the boundary faces where normalization breaks (typically exactly the named degeneracies $\gamma = 0$ or $\delta = 0$).

Proposition 31.10 (Cover/atlas universality via compact normalization within the certified class)**.** *Suppose the admissible parameter space Θ_{raw} admits a continuous normalization map*

$$\mathcal{N} : \Theta_{\text{raw}} \to \Theta_{\text{norm}}$$

into a compact set Θ_{norm} such that:

- *the pipeline predicates (tightness, Doeblin positivity, and the RC1 automaton) are invariant under the normalization, and*
- *the robust margins (γ, δ) and the paired robust radius $\boldsymbol{\varepsilon}^*$ can be chosen as functions of $\mathcal{N}(\theta)$.*

If Θ_{norm} admits a good-global contraction certificate within the certified class on the compact normalized family Θ_{norm} (for example by Proposition 31.4 or by closing region-mode coverage as in Proposition 31.5), then the corresponding universality conclusion holds for all parameters $\theta \in \Theta_{\text{raw}}$ whose normalization is defined. Moreover, the complement is explicitly contained in the normalization boundary, which is a subset of the named degeneracy loci.

Proof. By hypothesis, the decision predicates and the robustness radii depend only on the normalized parameters. Therefore the uniform contraction conclusion on Θ_{norm} pulls back to Θ_{raw} wherever $\mathcal{N}$ is defined. The only possible failure modes are points where normalization is undefined or discontinuous, which in positive-template settings correspond to entries collapsing to zero or to tightness degeneracies; these are contained in the exceptional set described earlier. □

Remark 31.11 (What remains to claim a truly "global" theorem)**.** *Proposition 31.10 turns the remaining work into two concrete tasks.*

- *Verify that your chosen normalization $\mathcal{N}$ preserves the admissible class and the pipeline invariants.*
- *Prove (or certify) good-global contraction universality within the certified class on the compact normalized space Θ_{norm}.*

The second task is exactly the compact problem solved by the chamberwise + region-mode machinery. The first task is model-dependent but typically elementary (it is an algebraic check once the normalization is fixed).

31.9 Validated global margins from finite sampling

Theorem 31.2 becomes genuinely global once one has uniform lower bounds γ_0, δ_0 on the margins across Θ. When Θ is a parameter family, a standard way to obtain such bounds is to combine a finite net with a conservative Lipschitz envelope. This subsection records a clean "verification lemma" that can be suitable for citation and implemented by a third party.

Definition 31.12 (Net mesh)**.** *Let $\Theta \subset \mathbb{R}^d$ be compact and let $\|\cdot\|$ be a fixed norm. A finite set $\mathcal{N} \subset \Theta$ is an h-net if for every $\theta \in \Theta$ there exists $\theta_i \in \mathcal{N}$ with $\|\theta - \theta_i\| \le h$.*

Lemma 31.13 (Certified minima from a Lipschitz envelope)**.** *Let $f : \Theta \to \mathbb{R}$ be Lipschitz with constant L_f with respect to $\|\cdot\|$. Let $\mathcal{N}$ be an h-net in Θ. Then*

$$\min_{\theta\in\Theta} f(\theta) \;\ge\; \min_{\theta_i\in\mathcal{N}} f(\theta_i) \;-\; L_f\, h.$$

In particular, if the right-hand side is strictly positive then $f(\theta) > 0$ for all $\theta \in \Theta$.

Proof. Fix $\theta \in \Theta$ and pick $\theta_i \in \mathcal{N}$ with $\|\theta - \theta_i\| \le h$. Lipschitz continuity gives $f(\theta) \ge f(\theta_i) - L_f\, h$. Taking the minimum over $\theta \in \Theta$ and then the minimum over $\theta_i \in \mathcal{N}$ yields the claim. □

Proposition 31.14 (Uniform margin verification by net + Lipschitz bounds)**.** *Assume Θ is compact. Fix a candidate tight SCC H and a candidate witness word w and assume these are constant on Θ. Assume:*

- *the tightness gap function $\gamma(\theta)$ is Lipschitz with constant L_γ,*
- *the Doeblin margin function $\delta_w(\theta) := \min(P_w(\theta))$ is Lipschitz with constant L_δ,*
- *RC1 holds for (H, w) (equivalently: the avoidance automaton has no directed cycle). In canonical mode, this may be upgraded to MB–RC1 for the canonical family $\mathcal{B}(H)$ (Section 13).*

Let $\mathcal{N}$ be an h-net in Θ. Define certified lower bounds

$$\underline{\gamma}_0 := \min_{\theta_i\in\mathcal{N}} \gamma(\theta_i) - L_\gamma h, \qquad \underline{\delta}_0 := \min_{\theta_i\in\mathcal{N}} \delta_w(\theta_i) - L_\delta h.$$

If $\underline{\gamma}_0 > 0$ and $\underline{\delta}_0 > 0$, then Θ admits uniform robust margins in the sense of Definition 31.1, hence Theorem 31.2 applies with $\gamma_0 = \underline{\gamma}_0$ and $\delta_0 = \underline{\delta}_0$.

Lemma 31.15 (RC1 is chamberwise constant)**.** *Fix a candidate tight SCC H and a candidate witness word w. Let $\mathcal{A}(H, w)$ denote the tight-language avoidance automaton used to encode RC1. If H and w are fixed on Θ, then $\mathcal{A}(H, w)$ is fixed on Θ. Consequently, either RC1 holds for all $\theta \in \Theta$ or RC1 fails for all $\theta \in \Theta$. In particular, RC1 can be verified at a single parameter value once the chamber is fixed.*

Proof. RC1 is equivalent to the absence of a directed cycle in $\mathcal{A}(H, w)$. When H and w are fixed, the transition structure of $\mathcal{A}(H, w)$ does not depend on θ. Thus the cycle predicate is independent of θ, proving the claim. □

Proof of Proposition 31.14. Lemma 31.13 applied to $f = \gamma$ and $f = \delta_w$ yields $\gamma(\theta) \ge \underline{\gamma}_0$ and $\delta_w(\theta) \ge \underline{\delta}_0$ for all $\theta \in \Theta$. RC1 supplies the recurrence-forcing item. Thus Definition 31.1 holds with the certified bounds. □

Remark 31.16 (How to obtain conservative Lipschitz constants)**.** *The proposition treats L_γ and L_δ as inputs. In applications, one bounds these constants from the parameterization. For example, if each template map $\theta \mapsto A_k(\theta)$ is Lipschitz in operator $\|\cdot\|_\infty$ with constant L_A and $\max_{\theta\in\Theta}\|A_k(\theta)\|_\infty \le M$, then for a fixed word w of length L one has the conservative envelope*

$$\|P_w(\theta) - P_w(\theta')\|_\infty \;\le\; L\,L_A\,M^{L-1}\,\|\theta - \theta'\|,$$

so one may take $L_\delta \le L\,L_A\,M^{L-1}$. For γ, a corresponding bound can be obtained once the reduced-cost computation is expressed as a finite composition of linear maps and maxima/minima whose coefficients are bounded on Θ; the pipeline may record a conservative L_γ as part of the calibration contract. At the artifact level, the parameter-family descriptor may declare the needed moduli explicitly via the `net_certification_contract` block in `theta_family_schema.json`.

31.10 Explicit Lipschitz constants for the margins on a fixed witness chamber

On a chamber where the witness data are frozen, one can make the verification inputs L_γ and L_δ completely explicit. The purpose of this subsection is to record conservative formulas that can be cited without invoking any hidden regularity assumptions.

Definition 31.17 (Chamberwise frozen witness object)**.** *Let $\Theta \subset \mathbb{R}^d$ be a compact parameter region. We say that Θ lies in a* frozen witness chamber *if there exist:*

- *a fixed calibrated pair (u, λ_*),*
- *a fixed tight edge set $H \subseteq E$ defined by the sign test for c_u using (u, λ_*),*
- *a fixed Doeblin witness word w of length L,*

such that all certificates on Θ are evaluated using this same data. In particular, the margin functions are $\gamma(\theta) = \min\{c_u(\theta, e) : e \in E \setminus H\}$ and $\delta_w(\theta) = \min(P_w(\theta))$ with H and w fixed.

Lemma 31.18 (Lipschitz constant for the tightness gap under cost parameterization)**.** *Assume Θ lies in a frozen witness chamber in the sense of Definition 31.17. Assume the cost map $\theta \mapsto c(\theta) \in \mathbb{R}^E$ is Lipschitz in $\|\cdot\|_\infty$ with constant L_c, i.e. $\|c(\theta) - c(\theta')\|_\infty \le L_c\,\|\theta - \theta'\|$ for all $\theta, \theta' \in \Theta$. Keep (u, λ_*) fixed. Then every reduced cost $c_u(\theta, e)$ is Lipschitz with constant L_c, and the tightness gap $\gamma(\theta) = \min_{e\in E\setminus H} c_u(\theta, e)$ is Lipschitz with constant*

$$L_\gamma \le L_c.$$

Proof. For fixed (u, λ_*), the map $c \mapsto c_u(e) = c(e) - \lambda_* + u(\mathrm{tail}(e)) - u(\mathrm{head}(e))$ is affine with Lipschitz constant 1 in $\|\cdot\|_\infty$. Hence $|c_u(\theta, e) - c_u(\theta', e)| \le \|c(\theta) - c(\theta')\|_\infty \le L_c\,\|\theta - \theta'\|$. The minimum of finitely many L_c-Lipschitz functions is L_c-Lipschitz, which yields the claim for γ. □

Lemma 31.19 (Lipschitz constant for the Doeblin margin under template parameterization)**.** *Assume Θ lies in a frozen witness chamber. Fix a witness word $w = w_1 \cdots w_L$. Assume each template map $\theta \mapsto A^{(k)}(\theta)$ is Lipschitz in entrywise $\|\cdot\|_{\infty,\mathrm{ent}}$ with constant L_A, and assume a uniform bound $\max_{\theta\in\Theta}\max_{k,i,j} A^{(k)}_{ij}(\theta) \le M$. Let $P_w(\theta) = A^{(w_L)}(\theta)\cdots A^{(w_1)}(\theta)$ and $\delta_w(\theta) = \min_{i,j}(P_w(\theta))_{ij}$. Then δ_w is Lipschitz with constant*

$$L_\delta \le L\,L_A\,M^{L-1}.$$

Proof. The telescoping identity gives $P_w(\theta) - P_w(\theta') = \sum_{t=1}^{L} A^{(w_L)}(\theta) \cdots A^{(w_{t+1})}(\theta)\big(A^{(w_t)}(\theta) - A^{(w_t)}(\theta')\big)A^{(w_{t-1})}(\theta') \cdots A^{(w_1)}(\theta')$ with the obvious conventions at the ends. Taking entrywise $\|\cdot\|_{\infty,\mathrm{ent}}$ and using the uniform bound M yields $\|P_w(\theta) - P_w(\theta')\|_{\infty,\mathrm{ent}} \le \sum_{t=1}^{L} M^{L-1}\,\|A^{(w_t)}(\theta) - A^{(w_t)}(\theta')\|_{\infty,\mathrm{ent}} \le L\,M^{L-1}\,L_A\,\|\theta - \theta'\|$. Since δ_w is the minimum of the finitely many entries of P_w, it is 1-Lipschitz as a function of the entry vector, hence it inherits the same constant. $\square$

Remark 31.20 (Verified bounds without smoothness)**.** *The formulas in Lemmas 31.18 and 31.19 only use finite-dimensional Lipschitz envelopes and do not require differentiability. When a parameter map is only piecewise-Lipschitz, one may apply the same verification locally on each piece and then use the chamberwise cover mechanism (Algorithm 9).*

31.11 End-to-end artifact verification (report, atlas, and witnesses)

The net+Lipschitz verification lemmas above certify *numerical* margin lower bounds. Separately, the artifact layer is designed so that a third party can verify the global decision *without rerunning* the full internal pipeline. The key point is that every chamber label ultimately reduces to a finite pointwise frontier report, and pointwise frontier reports are finitely verifiable (Algorithm 8).

Lemma 31.21 (Frontier-atlas reports are finitely verifiable)**.** *Given a `GlobalUniversalityReport` together with its embedded `FrontierAtlasReport` and the list of referenced pointwise payloads (robust certificates and obstruction witnesses), there is a finite procedure that checks:*

- *the JSON schema validity of the report and atlas objects,*
- *that every referenced certificate/witness file exists and validates against its schema,*
- *that for every referenced payload, its embedded frontier report passes* VerifyFrontierReport*,*
- *and that the atlas index is consistent with the set of verified payload outcomes.*

In particular, on fixed-support families the atlas can be treated as an end-to-end witness object for the global decision.

Algorithm 10 VerifyGlobalReport (batch-check a global-on-Θ decision)

Require: GlobalUniversalityReport $\mathcal{R}$ with embedded FrontierAtlasReport $\mathcal{A}$, and a directory of referenced payload files

Ensure: PASS/FAIL with failures

1: **for** each referenced payload id/file in $\mathcal{A}$.`index` **do**
2: Load payload object W (robust certificate or obstruction witness)
3: Validate W against its schema; else FAIL
4: Extract embedded frontier report R_W from W
5: Run VerifyFrontierReport(R_W, W) (Algorithm 8); else FAIL
6: **end for**
7: For each chamber entry K in $\mathcal{A}$.`chambers`, if K.`chamber_invariance.cover_refs` is nonempty, check that every referenced payload was verified and has local invariance radius $\geq$ K.`chamber_invariance.eps_dec`; else FAIL
8: Check that $\mathcal{A}$.`stratification.labels_present` equals the set of chamber outcome labels and that $\mathcal{A}$.`stratification.chambers_by_label` groups exactly the chamber IDs by label; else FAIL
9: If $\mathcal{A}$.`coverage.uncovered_is_empty` is true, check that all chambers have label `contraction` and that $\mathcal{A}$.`loci` is empty; else FAIL
10: Check atlas chamber labels and loci are consistent with the verified payload outcomes; else FAIL
11: **return** PASS if no check fails; otherwise return FAIL with failures

32 A Global Cover/Atlas Decider on Compact Normalized Families

The robust-certificate contract makes every local claim (tight-core structure, Doeblin persistence, recurrence mode, and rate pack) an *open* property with an explicit stability radius. This section turns that openness into a *global* statement on a fixed compact normalized family Θ within the certified class. The outcome is a finite, witness-producing decision procedure: either the family is certified globally by a single paired robustness radius and uniform rate pack, or the procedure returns an explicit obstruction/degeneracy report identifying where and why a global certificate cannot hold.

32.1 Setup and the property being decided

Fix a compact parameter set Θ together with a normalization guaranteeing uniform bounds on all template entries and reduced-data constants appearing in the robust certificate. For each $\theta \in \Theta$, the instance induces reduced graph data and the certificate pipeline returns exactly one of the outcomes in Section 20.

Definition 32.1 (Good-global contraction universality on Θ)**.** *We say* good-global contraction universality *holds on* Θ *if there exist* $\boldsymbol{\varepsilon}_{\text{global}} = (\varepsilon_{c,\text{global}}, \varepsilon_{A,\text{global}})$ *and a uniform rate pack* $(\tau_{\text{global}}, C_{\text{global}}, \kappa_{\text{global}})$ such that for every $\theta \in \Theta$ *the pipeline returns a contraction certificate and, moreover, the certificate remains valid under all perturbations satisfying* $\|\Delta c\| \leq \varepsilon_{c,\text{global}}$ *and* $\|\Delta A\| \leq \varepsilon_{A,\text{global}}$ *(in the chosen perturbation model), yielding uniform exponential projective contraction with the stated rate pack. We also write the conservative scalar floor* $\varepsilon_{\min,\text{global}} := \min\{\varepsilon_{c,\text{global}}, \varepsilon_{A,\text{global}}\}$ *when a single-radius parameter-space cover statement is sufficient.*

Definition 32.2 (Boundary-global regime atlas on Θ)**.** *A boundary-global regime atlas on Θ is a finite, witness-producing report that either certifies Definition 32.1, or else exhibits explicit obstruction witnesses and/or a nonempty uncovered set together with diagnostics placing it near the named degeneracy loci $\{\gamma = 0\} \cup \{\delta = 0\} \cup \{\text{recurrence gate failure}\}$.*

The atlas viewpoint is the honest global completion. It never asserts uniform contraction on a region that intersects the degeneracy boundary, and it never leaves the exceptional set implicit.

32.2 Decision procedure and completeness

The proof mechanism has two ingredients already established:

- the robust certificate predicates are open with explicit stability radii (Sections 25, 26, 27, 28);
- the region-cover route: finitely many certificates cover Θ by neighborhoods of radius $\varepsilon^*_{\min}(\theta_i)$ (Section 30).

We now combine these into a theorem whose conclusion is *algorithmic and dichotomic.*

Theorem 32.3 (Global cover/atlas decider on Θ)**.** *Assume Θ is compact and that the normalization chosen in Section 2 provides uniform bounds sufficient to evaluate every constant appearing in the robust certificate fields. Run the region-mode procedure of Section 30 with adaptive refinement. The stopping criterion is: either the certified neighborhoods close coverage (uncovered set empty), or the remaining uncovered set is certified to lie inside the declared degeneracy neighborhood; in both cases the* GlobalUniversalityReport *is the output object. Then the procedure terminates and outputs a* GlobalUniversalityReport *Moreover, the report embeds a* FrontierAtlasReport *under the field* `frontier_atlas` *(Section 22), so the global outcome is available both as a full audit artifact and as a single frontier-compression object. The full report plus its atlas can be batch-verified end-to-end by validating referenced payloads and running* VerifyFrontierReport *chamberwise; see Section 31.11. The global report has exactly one of the following outcomes.*

he certified class) *The uncovered set is empty. Then Definition 32.1 holds on Θ with $\boldsymbol{\varepsilon}_{\text{global}} := \big(\min_i \varepsilon^*_c(\theta_i),\ \min_i \varepsilon^*_A(\theta_i)\big)$ and a uniform rate pack obtained by taking conservative maxima/minima over the finitely many center certificates.*

dary-global atlas) *The output contains either*

- *a finite obstruction witness (one of OG1–OG4) at some sampled parameter, or*
- *a nonempty uncovered set together with diagnostics placing it inside a neighborhood of the explicit degeneracy surfaces $\{\gamma = 0\} \cup \{\delta = 0\} \cup \{\text{recurrence gate failure}\}$ as defined in Section 31.*

In this case, no choice of a single paired radius can certify the entire region by the robust-certificate method without entering the reported degeneracy neighborhood.

Proof. For each parameter value θ, the robust certificate defines an explicit paired radius $\boldsymbol{\varepsilon}^*(\theta)$ whenever the certificate branch conditions have positive margins. By the openness lemmas (Robust Certificate Contract through Recurrence Stability), the same certificate remains valid throughout the corresponding perturbation neighborhood. Compactness of Θ implies there exists a finite subcover of any open cover. The region-mode algorithm either constructs such a finite subcover explicitly (yielding empty uncovered set), or it refines until any remaining uncovered region lies within the declared degeneracy neighborhood, since outside that neighborhood the margins are bounded away from zero by the Lipschitz-net verification route of Lemma 31.13. Obstruction witnesses OG1–OG4 are finite and explicit by the pipeline contract. The report combines these outputs. □

Remark 32.4 (Certificate object for downstream use)**.** *When the outcome is certified, the global information needed by a third party is typically smaller than the full region report. The file* `global_universality_certificate_schema.json` *records a minimal object containing:* $\varepsilon_{\mathrm{global}}$*,* γ_0*,* δ_0*, a witness record (uniform or chamberwise), and the uniform rate pack. This is the natural carry-forward artifact for later proofs or numerical experiments.*

Corollary 32.5 (When the decider must certify: degeneracy avoidance on finitely many witness chambers)**.** *Assume* Θ *is compact and is covered by finitely many frozen witness chambers in the sense of Definition 31.17. If on each chamber the degeneracy sets of Definition 31.3 are empty (equivalently: the verified lower bounds in Proposition 31.14 are strictly positive and the chosen recurrence gate holds on that chamber (RC1 in single-block mode; MB–RC1 in canonical multi-block mode)), then the region-mode procedure returns the certified outcome of Theorem 32.3. In particular, good-global contraction universality holds on* Θ *with an explicit uniform paired radius* $\varepsilon_{\mathrm{global}} := \min_{\textit{chambers}\ \Omega} \varepsilon_{\mathrm{global}}(\Omega)$ *(componentwise minimum) and a uniform rate pack.*

Proof. On each frozen witness chamber Ω, the emptiness of degeneracy sets implies uniform robust margins by Proposition 31.4, and the explicit net+Lipschitz route supplies certified lower bounds when desired (Proposition 31.14). Thus Theorem 31.2 applies chamberwise and produces an explicit chamberwise paired radius $\varepsilon_{\mathrm{global}}(\Omega)$. Finiteness of the chamber cover gives a positive global componentwise minimum, and compactness yields a finite cover by robust certificate neighborhoods. Therefore the region-mode decider closes coverage and outputs the certified case. □

32.3 Consequences for decidability islands

Theorem 32.3 converts a qualitative "robustness in a neighborhood" statement into a concrete, checkable conclusion for parameter families. In particular, it yields a sharp *decidable island* statement for robust switching stability: on the certified positive/tight/Doeblin subclass supporting the pipeline, the question "does this compact normalized family admit a uniform robust contraction mechanism?" reduces to a finite cover computation plus explicit degeneracy diagnostics.

Remark 32.6 (What this does and does not claim)**.** *The decider above is a* family-level *completeness statement for the certified class. It does not claim that good-global contraction universality holds for all possible parameter families, nor that the general joint spectral radius decision problem is decidable. Instead, it provides an explicit frontier: inside the certified class and under normalization, the program decides the outcome by producing either a global certificate or explicit obstruction/degeneracy data.*

32.4 Artifact-level completion: report, certificate, and paired radii

The mathematics above naturally produces two levels of machine-checkable output. The intent is that an independent reader can: (i) validate the global conclusion without rerunning the full internal pipeline, and (ii) carry forward the minimal global data needed to support later proofs.

Paired radii and the conservative scalar floor

The robust layer distinguishes perturbations of costs and perturbations of template entries. Accordingly, the canonical robustness radius is a pair $\boldsymbol{\varepsilon} = (\varepsilon_c, \varepsilon_A)$ with the meaning $\|\Delta c\| \le \varepsilon_c$ and $\|\Delta A\| \le \varepsilon_A$. For convenience, we also record the conservative scalar floor $\varepsilon_{\min} := \min\{\varepsilon_c, \varepsilon_A\}$. The scalar floor is sufficient for parameter-space cover arguments; the pair is the natural object for reproducible robustness claims.

Report versus certificate

Artifact	Role	When produced
`GlobalUniversalityReport`	Full region-level audit: cover data, diagnostics, and optional proofs	Always (certified or boundary)
`GlobalUniversalityCertificate`	Compressed carry-forward object: global bounds + witnesses + rate pack	Only when uncovered set is empty

Headline artifact: the frontier atlas report. In addition to the full region report, we also treat a third object as the primary human-facing summary: the `FrontierAtlasReport`. This object compresses the full report into: (i) a chamber list labeled by minimal-gate outcomes (`contraction` or OG1–OG4, plus `near_degeneracy`), (ii) explicit boundary loci, and (iii) an index of referenced certificates and witnesses. It is intended as the headline artifact for operational "frontier mapping" claims.

Both artifacts are explicitly specified by JSON Schemas:

The report schema supports the paired-radius form by allowing `epsilon_global` and `epsilon_star` to be either a scalar or an object with fields `eps_c_star`, `eps_A_star`, and (optionally) `eps_min`.

Compression map: report $\to$ certificate

In the certified case, the report contains more information than downstream use typically requires. The compression map discards only: coverage bookkeeping (net points and intermediate certificate IDs), optional Lipschitz proof blocks, and free-text diagnostics. It retains:

- global margins (γ_0, δ_0),
- the paired global robustness radius $\boldsymbol{\varepsilon}_{\text{global}}$,
- the witness payload (uniform or chamberwise),
- and the uniform rate pack.

This is the exact data needed to reuse the global conclusion as an input to later arguments.

Compression map: report $\to$ frontier atlas

The frontier atlas report is a second compression, aimed at producing a single headline object. It retains chamber labels, minimal-gate outcomes, boundary loci, and an index of witness references.

Algorithm 11 CompressToFrontierAtlas (report-level frontier compression)

Require: GlobalUniversalityReport $\mathcal{R}$

Ensure: FrontierAtlasReport $\mathcal{A}$

1: Set $\mathcal{A}$.`theta_region` $\leftarrow$ $\mathcal{R}$.`theta_region`

2: Set $\mathcal{A}$.`summary.good_global_contraction_universality` $\leftarrow$ $\mathcal{R}$.`conclusion.global_contraction_universality`

3: Initialize chamber list from $\mathcal{R}$.`verification_method.chambers` when present

4: For each chamber/ball label, attach evidence:
   if covered, store certificate id(s) and outcome `contraction`
   if obstruction found, store witness ref and outcome OG1–OG4
   if near degeneracy, store degeneracy types and outcome `near_degeneracy`

5: Set $\mathcal{A}$.`loci` from $\mathcal{R}$.`coverage.uncovered.near_degeneracy` and any obstruction labels

6: Build $\mathcal{A}$.`index` by collecting referenced certificate IDs and witness files

7: Emit a short headline and recommendation string based on whether the uncovered set is empty

Schema validation

The schema files are included so that the report and certificate objects can be validated independently with any standards-compliant JSON Schema validator.

Beyond schema checks, the intended verification route is witness-based. A third party can batch-check the entire global decision by validating every referenced pointwise payload and running VerifyFrontierReport on each embedded frontier report. This is recorded as VerifyGlobalReport in Section 31.11.

Machine description of the target family Θ

The global decider is a statement on a *region*, so the artifacts include a separate, explicit descriptor for the parameter family.

When one invokes the "global-on-Θ by net + Lipschitz envelope" route (Proposition 31.14), the descriptor also makes the required continuity inputs explicit via the `net_certification_contract` block. This block declares the parameter norm and conservative moduli (L_c, L_A, M) used to certify uniform positive margins from finite sampling.

- `global_universality_report_ThetaCan_certified.json` (certified-good-global format example)
- `global_universality_certificate_ThetaCan.json` (compressed carry-forward certificate)
- `global_universality_report_ThetaCan_atlas.json` (boundary-global atlas format example)
- `frontier_atlas_report_ThetaCan_certified.json` (frontier atlas compression example)
- `frontier_atlas_report_ThetaCan_atlas.json` (frontier atlas compression example)

33 Sharpness and Realizability of the Obstruction Taxonomy

This section records explicit, finite examples showing that each obstruction class OG1–OG4 is *genuinely necessary*. We also give a concrete "good regime" example where OG1–OG4 do not occur and the forcing mechanism (Sections 8 and 10) yields a nonvacuous unconditional contraction rate.

Throughout, we present examples at the reduced-graph/template level:

- vertices represent descriptor states,
- each directed edge $e : u \to v$ carries a nonnegative template matrix T_e,
- the canonical support sets $S(\cdot)$ are specified explicitly and checked by support propagation (Definition 11.1).

33.1 OG1 is realizable (no aperiodic tight SCC)

Let $V = \{a, b\}$ and let the tight graph have exactly the edges $a \to b$ and $b \to a$. This SCC has period 2, hence it is imprimitive. Set $d = 1$ and take $T_{a\to b} = T_{b\to a} = [1]$. Then every edge is tight, the unique SCC has period 2, and therefore OG1 holds. No aperiodic tight SCC exists, so no Doeblin/minorization regime can be entered.

33.2 OG2 is realizable (face-map inconsistency cycle)

Let $V = \{u, v\}$, $d = 2$. Define support sets

$$S(u) = \{1\}, \qquad S(v) = \{1\}.$$

Let the reduced graph contain both edges $u \to v$ and $v \to u$ (so it is an SCC). Assign the template on the edge $e : u \to v$ by

$$T_e = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}.$$

Then for $x \in \mathbb{R}^{S(u)}_{>0}$ we have $T_e x = (x_1, x_1)$, so coordinate 2 is activated. But $2 \notin S(v)$, hence the "no spurious activation" condition in Definition 11.1 fails. By Lemma 11.9, the edge e together with the marked indices $(i, j) = (2, 1)$ and any return path $v \rightsquigarrow u$ produces a finite inconsistency cycle witness. Thus OG2 is realizable.

33.3 OG3 is realizable (no Doeblin word on the active face)

We record two standard sharp shapes corresponding to the two OG3 witness types.

33.3.1 OG3(cut) realizable: block-triangular support

Let $V = \{h\}$ (one-vertex SCC), $d = 2$, $S(h) = \{1, 2\}$. Let the unique edge be a self-loop $e : h \to h$ with

$$T_e = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Then $S(\cdot)$ is a face map and $S(H) = \{1, 2\}$. However, there is no arrow $2 \to 1$ in the coordinate digraph $\Gamma(H)$ (Definition 11.4), so $\Gamma(H)$ is not strongly connected. Equivalently, letting $A = \{1\}$ and $B = \{2\}$, we have $(T_e)_{1,2} = 0$ and hence $(B(w))_{1,2} = 0$ for every word w. Thus no word can be strictly positive on $S(H) \times S(H)$ and OG3 holds with a cut witness (Lemma 11.10).

33.3.2 OG3(product-automaton) realizable: permutation-only regime

Let $V = \{h\}$, $d = 2$, $S(h) = \{1, 2\}$. Let the allowed generator templates be the two permutation matrices

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad Q = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and let the reduced graph have two self-loop edges at h labeled by P and Q. Then $\Gamma(H)$ is strongly connected (each coordinate reaches the other under Q), but every product is still a permutation matrix, so no word yields strict positivity on $S(H) \times S(H)$. This is exactly the "strongly connected but not Doeblin" counter-shape in Remark 11.7, and the Boolean-pattern product automaton returns a finite non-reachability certificate. Thus OG3 is realizable even in the presence of strong connectivity.

33.4 OG4 is realizable (Doeblin exists but is avoidable)

We now realize OG4 in the regime where OG1–OG3 fail. Let $V = \{h, x, y\}$ and consider the tight SCC H supported on these vertices. Take $d = 2$ and $S(v) = \{1, 2\}$ for all v. Let there be edges

$$h \to x, \quad x \to h, \quad x \to y, \quad y \to x.$$

This SCC is aperiodic (it contains a 2-cycle $h \to x \to h$ and a 3-cycle $x \to y \to x \to h \to x$). Assign templates:

$$T_{h \to x} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{(strictly positive)}, \qquad T_{x \to h} = I, \qquad T_{x \to y} = Q, \qquad T_{y \to x} = Q,$$

where I is the identity and Q is the swap matrix from the preceding example. Then the single-edge word $w = (h \to x)$ is a Doeblin word (already strictly positive), so OG3 fails. However, the periodic tight path alternating $x \to y \to x \to y \to \cdots$ avoids the vertex h entirely, so it avoids w entirely. In particular, the avoidance graph $\mathcal{A}_{\neg w}(H)$ has a directed cycle and OG4(H, w) holds (Definition 16.1).

This shows that even in the "good" structural regime where a Doeblin word exists, bounded-gap recurrence of that *chosen* word is not automatic.

33.5 Nonvacuous good regime: OG1–OG4 fail and forcing succeeds

Finally, we give a compact example where the forcing mechanism is nontrivial. Let $V = \{h, x, y\}$ and edges

$$h \to x, \quad x \to h, \quad x \to y, \quad y \to h.$$

Assume these are the only outgoing edges from h (so h has outdegree 1). The SCC H on $\{h, x, y\}$ is strongly connected and aperiodic: there is a cycle of length 2 ($h \to x \to h$) and a cycle of length 3 ($h \to x \to y \to h$). Let $d = 2$ and $S(v) = \{1, 2\}$ for all v. Assign templates

$$T_{h \to x} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \qquad T_{x \to h} = I, \qquad T_{x \to y} = I, \qquad T_{y \to h} = I.$$

Then $w = (h \to x)$ is a Doeblin word. Moreover, because h has a unique outgoing edge, *every* infinite path in H contains the edge $h \to x$ with gaps at most 3 (in fact at most the maximum return time to h). Equivalently, the avoidance graph $\mathcal{A}_{\neg w}(H)$ is acyclic, so RC1 holds and OG4 fails. Therefore the forcing dichotomy applies (Corollary 10.5), and Theorem 17.8 yields a uniform exponential contraction rate on all tight minimizers supported in H.

Remark 33.1 (What this demonstrates)**.** *This example is intentionally simple: the Doeblin block is a single positive generator and recurrence is forced by a finite outdegree constraint. The point is not complexity but* nonvacuity*: the "OG1–OG4 fail $\Rightarrow$ contraction" regime is populated, and the forcing lemmas deliver explicit rates from finite witnesses.*

34 Minimality and sharpness: why the boundary is tight

This section records short "edge-case" constructions that explain why the hypotheses and obstruction gates in the boundary theorem cannot be substantially weakened without changing the conclusion. The goal is not to catalog counterexamples, but to isolate the minimal mechanisms that force the regime split.

34.1 Why tight-core reduction is not optional

The calibration potential ϕ and the induced tight graph $G^{(0)}$ are not merely a technical convenience. They are the device that makes the extremal problem finite and local. Without passing to tight edges, statements about minimizers are typically false if one reads them directly on the unreduced descriptor graph.

Remark 34.1 (A two-cycle tie shows why one must reduce)**.** *Consider a finite graph with two disjoint cycles γ_1 and γ_2 that have the same cycle mean λ_*. Assume there are additional edges of slightly higher mean that connect the cycles in both directions. Then minimizing measures may be supported on either cycle, or on mixtures that switch between the two cycles with sublinear frequency. On the unreduced graph there is no canonical "core" and no unique active face. After calibration, the tight subgraph separates the true minimizer-supporting structure from slack edges; the subsequent obstruction tests are meaningful only on this tight reduction.*

34.2 How contraction fails: realizable obstructions

The obstruction gates OG2 and OG3 are designed to be witness-producing and to capture genuinely different failure modes. Both occur in simple finite-template systems.

- **OG2 (face-map inconsistency) is a real obstruction.** It is possible for tight edges to force incompatible coordinate support requirements. Concretely, one can have two tight words u and v in the same tight SCC such that u maps mass from an active face into a strict subface, while v maps mass into a different strict subface, and the SCC allows alternating concatenations. The witness produced by Lemma `lem:og2-witness-cycle` is exactly the finite inconsistency cycle that certifies that no consistent face map exists.
- **OG3 (no Doeblin word) is strictly stronger than "$\Gamma(H)$ not strongly connected."** Even when the union support digraph $\Gamma(H)$ is strongly connected, the allowed face-restricted patterns may generate a semigroup that never contains an all-positive block. The clean counter-shape is the "permutation-pattern" family: if the admissible patterns on the face are permutation matrices that generate a transitive action, then $\Gamma(H)$ is strongly connected but every product remains a permutation pattern, so no Doeblin word exists. This is why the certificate used in the paper is the Boolean-pattern product-automaton test: Doeblin existence is equivalent to reachability of the all-ones pattern state.

34.3 When imprimitive trapping is the right conclusion

The regime split treats aperiodicity (period 1) and imprimitivity (period $p > 1$) differently because the projective geometry is different. In an imprimitive tight SCC, the index set splits into cyclic classes and products respect that cyclic structure. Even with strong tightness and perfect calibration, one should not expect a single stationary projective section on the full face.

Remark 34.2 (Imprimitive tight cores can force cycling)**.** *Let H be a tight SCC with period $p > 1$. Then there is a cyclic decomposition of states into classes $C_0, \dots, C_{p-1}$ such that every edge advances the class modulo p. Along an admissible tight trajectory, the support of mass on the active face can rotate among these classes. This produces a genuine periodic trapping phenomenon at the level of projective directions. Accordingly, the boundary theorem outputs "imprimitive trapping" (periodic regime) rather than uniform contraction on the full face.*

Taken together, these examples explain the sharpness of the theorem chain: the tight-core reduction is needed to make minimizers finite and local; OG2 and OG3 isolate distinct algebraic failures on the active face; and the imprimitive branch is the correct endpoint when the tight core has period > 1.

35 Parameter Chambers and Generic Cycle Selection

This section formalizes what is generic in linear parameter families in a literal, checkable sense: uniqueness of the minimizing cycle mean off a finite hyperplane arrangement, together with a finite chamber decomposition of parameter space on which the minimizing set is constant. No claim of "generic aperiodicity" is made, since periodic/aperiodic is a combinatorial property of the selected critical SCC.

35.1 Linear parameter model and hyperplane arrangement

Fix a finite directed graph $G = (V, E)$ (after reduction) and a feature map $\psi : E \to \mathbb{Z}^k$. For $\theta \in \mathbb{R}^k$ define edge costs

$$c_\theta(e) = \langle \theta, \psi(e) \rangle,$$

and for a directed cycle γ define its mean cost

$$\mu_\theta(\gamma) := \frac{1}{|\gamma|} \sum_{e \in \gamma} c_\theta(e).$$

Let $\mathcal{C}$ be the finite set of simple directed cycles in G and let

$$\lambda_*(\theta) := \min_{\gamma \in \mathcal{C}} \mu_\theta(\gamma)$$

be the minimal cycle mean.

Definition 35.1 (Degeneracy hyperplanes)**.** *For distinct cycles $\gamma \neq \gamma'$ define the affine functional*

$$F_{\gamma,\gamma'}(\theta) := \mu_\theta(\gamma) - \mu_\theta(\gamma').$$

The degeneracy arrangement *is the finite union of hyperplanes*

$$\mathcal{D} := \bigcup_{\gamma \neq \gamma'} \{\theta : \; F_{\gamma,\gamma'}(\theta) = 0\}.$$

Lemma 35.2 (Unique minimizing cycle off $\mathcal{D}$)**.** *If $\theta \notin \mathcal{D}$, then there exists a unique cycle $\gamma_*(\theta) \in \mathcal{C}$ with $\mu_\theta(\gamma_*(\theta)) = \lambda_*(\theta)$.*

Proof. If two distinct cycles both attained the minimum, then $F_{\gamma,\gamma'}(\theta) = 0$ for that pair, so $\theta \in \mathcal{D}$. □

35.2 Chamber decomposition and critical SCC stability

Definition 35.3 (Chambers)**.** *A* parameter chamber *is a connected component of* $\mathbb{R}^k \setminus \mathcal{D}$*.*

Proposition 35.4 (Finite chamber structure)**.** $\mathbb{R}^k \setminus \mathcal{D}$ *is a finite union of open polyhedral chambers. On each chamber, the minimizing cycle* $\gamma_*(\theta)$ *is constant.*

Proof. $\mathcal{D}$ is a finite arrangement of affine hyperplanes, so its complement has finitely many connected components, each described by a finite system of strict linear inequalities $F_{\gamma,\gamma'}(\theta) \lessgtr 0$. Within a fixed component, the sign pattern of all $F_{\gamma,\gamma'}$ is constant, hence the unique minimizer γ_* is constant by Lemma 35.2. □

Corollary 35.5 (Computable periodic/aperiodic regime on chambers)**.** *Fix a chamber* $\mathcal{U}$ *and let* γ_* *be its minimizing cycle. Then the essential critical SCC of the tight graph is the SCC generated by* γ_* *(in particular, it is periodic). More generally, on lower-dimensional strata* $\theta \in \mathcal{D}$ *where multiple cycles tie, the tight critical subgraph may contain multiple cycles. It may be periodic or aperiodic depending on the gcd of cycle lengths in the resulting critical SCC. All of these properties are decidable from the finite tight geometry.*

Theorem 35.6 (Boundary type is constant on chambers)**.** *Fix the graph* G *and feature map* ψ *as above, and consider the induced linear cost family* c_θ*. For each chamber* $\mathcal{U}$ *of* $\mathbb{R}^k \setminus \mathcal{D}$*, the following data are constant over* $\theta \in \mathcal{U}$*:*

- *the minimizing cycle* γ_* *and minimal mean* $\lambda_*(\theta)$ *(hence the critical/tight SCC);*
- *the periodic/aperiodic regime of the essential critical SCC;*
- *the boundary classification output of the Tight-Core Boundary Theorem on that SCC (imprimitive trapping, or in the aperiodic case: certificate/Doeblin/contraction versus OG2/OG3 obstructions, together with the associated witness objects).*

Consequently, the boundary type can change only when crossing the degeneracy arrangement $\mathcal{D}$*.*

Proof. On a chamber $\mathcal{U}$ the sign pattern of all cycle-mean differences $F_{\gamma,\gamma'}$ is fixed, so the minimizer γ_* is constant by Proposition 35.4. In the unique-minimizer case, the critical/tight subgraph is generated by the edges of γ_*, hence the essential critical SCC (and its periodicity) is constant. All subsequent boundary outputs in the Tight-Core Boundary Theorem are computed from finite graph data on the essential SCC and its induced tight geometry (certificate graph construction, SCC type, and the witness-producing obstruction tests), so they are constant once that finite geometry is fixed. Crossing a hyperplane in $\mathcal{D}$ is the only way the minimizer set can change, hence it is the only place where the induced tight/critical geometry and boundary type can change. □

Edge cases on strata $\theta \in \mathcal{D}$. On the degeneracy arrangement, ties produce a critical subgraph with multiple cycles. The same finite pipeline applies, but the tight geometry must be formed from the tied minimizing set. For convenience, the most common tie patterns can be summarized as follows.

Situation on $\mathcal{D}$	What is decidable (and what to report)
Ties among cycles *within one* critical SCC	Compute the gcd of cycle lengths in that SCC. If the gcd is > 1, the regime is periodic (trapping). If the gcd is 1, the SCC is aperiodic and the Tight-Core Boundary Theorem applies (certificate/Doeblin/contraction or OG2/OG3 with witnesses).
Ties split across *multiple* critical SCCs	The tight/critical graph decomposes into SCCs. Run the boundary classification on each essential SCC and report the resulting regime/witnesses; the overall boundary type is the finite list of SCC outcomes.
Higher-codimension ties (many cycles)	Form the tight subgraph induced by the minimizing set and apply the same SCC + certificate/obstruction pipeline. The output remains finite and witness-producing.

Remark 35.7 (What this does (and does not) justify)**.** *In linear cost families, uniqueness of the minimizing cycle is generic in the precise sense "off $\mathcal{D}$." That generic situation typically yields a periodic critical SCC. Accordingly, any claim of "generic aperiodicity" would require additional structure beyond the linear cycle-mean model. The robust message supported here is different: the periodic/aperiodic and certificate/obstruction boundary is finitely computable from the tight geometry, and by Theorem 35.6 the boundary type is constant on chambers and can change only across $\mathcal{D}$.*

35.3 Certificate chambers: invariant balls with exported radii

The hyperplane arrangement discussion above explains how cycle selection changes in linear cost models. Independently of that special structure, the robust certificate program produces a second, operational notion of "chamber": a region piece on which the pipeline's *minimal-gate label* is provably constant on explicit neighborhoods. This is the chamber concept used by the exported `FrontierAtlasReport`.

Definition 35.8 (Certified ball and chamber label)**.** *Fix a parameter space Θ equipped with a perturbation metric and assume the fixed-support contract (Definition 3.1). A* certified ball *is a pair (θ, r) together with a verified pointwise frontier report whose* `local_invariance` *radius is at least r. Its* label *is the report's* `minimal_gate` *(*`contraction` *or one of OG1–OG4).*

Theorem 35.9 (Chamber stability on invariant balls with exported radii)**.** *Let Θ be a compact normalized family in the certified class satisfying the fixed-support contract. Let A be a* `FrontierAtlasReport` *produced by the global cover/atlas decider and* CompressToFrontierAtlas*. Fix a chamber Ω listed in A.*`chambers` *with outcome label L and chamberwise decision radius lower bound $\varepsilon_{\text{dec}} := A.$*`chambers`*$[\Omega]$.*`chamber_invariance.eps_dec`*. Then the following hold.*

- *For every certified center item referenced in* `chamber_invariance.cover_refs`*, the minimal-gate label equals L and is constant on the perturbation ball of radius at least ε_{dec} around its center.*
- *Consequently, on the union of these certified balls (the chamber's certified subcover), the minimal-gate label is everywhere equal to L.*
- *Any label change on Θ (under refinement or within a certified invariance neighborhood) is confined to neighborhoods of the degeneracy loci recorded in A.*`loci`*.*

Proof. By construction, `cover_refs` points to verified pointwise artifacts whose frontier reports carry a certified `local_invariance` radius. The compression algorithm defines $\varepsilon_{\mathrm{dec}}$ as the minimum of these local invariance radii across the chamber's certified subcover. Therefore each referenced center admits a neighborhood of size at least $\varepsilon_{\mathrm{dec}}$ on which the minimal-gate label is invariant and equal to L. The union-of-balls statement follows immediately. Finally, under the fixed-support contract, label changes can occur only where either tight-core structure changes or a certificate margin collapses. These possibilities are exactly the recorded degeneracy loci in the atlas. □

References

[1] V. D. Blondel and J. N. Tsitsiklis. The boundedness of all products of a pair of matrices is undecidable. *Systems & Control Letters* 41 (2000), 135–140.

[2] I. D. Morris. Mather sets for sequences of matrices and applications to the study of joint spectral radii. *Proceedings of the London Mathematical Society* 107 (2013), 121–150.

[3] V. Kozyakin. An annotated bibliography on the convergence of matrix products and the theory of joint/generalized spectral radius (JSRbib). Available at https://kozyakin.github.io/jsrbib/JSRbib.html. Accessed 2026-02-20.

[4] I. D. Morris and N. Sidorov. On a devil's staircase associated to the joint spectral radii of a family of pairs of matrices. *Journal of the European Mathematical Society*, 15(5):1747–1782, 2013. DOI: https://doi.org/10.4171/JEMS/402.

[5] H. Don, H. Zantema, and M. de Bondt. Slowly synchronizing automata with fixed alphabet size. *Information and Computation*, 279:104614, 2021. DOI: https://doi.org/10.1016/j.ic.2020.104614.

[6] A. N. Trahtman. Synchronization of some DFA. In *Theory and Applications of Models of Computation (TAMC 2007)*, Lecture Notes in Computer Science, vol. 4484, pages 234–243. Springer, 2007. DOI: https://doi.org/10.1007/978-3-540-72504-6_21.

[7] S. W. Neufeld. A diameter bound on the exponent of a primitive directed graph. *Linear Algebra and its Applications*, 245:27–47, 1996. DOI: https://doi.org/10.1016/0024-3795(94)00203-7.

[8] V. Yu. Protasov. The Barabanov norm is generically unique, simple, and easily computed. *SIAM Journal on Control and Optimization*, 60(4):2246–2267, 2022. DOI: https://doi.org/10.1137/21M1426821.

[9] É. Charlier and J. Honkala. The freeness problem over matrix semigroups and bounded languages. *Information and Computation*, 237:243–256, 2014. DOI: https://doi.org/10.1016/j.ic.2014.03.001.

[10] G. Birkhoff. Extensions of Jentzsch's theorem. *Transactions of the American Mathematical Society* 85 (1957), 219–227.

[11] P. J. Bushell. Hilbert's metric and positive contraction mappings in a Banach space. *Archive for Rational Mechanics and Analysis* 52 (1973), 330–338.

[12] B. Lemmens and R. Nussbaum. *Nonlinear Perron–Frobenius Theory*. Cambridge Tracts in Mathematics. Cambridge University Press, 2012.

[13] E. Seneta. *Non-negative Matrices and Markov Chains.* Springer Series in Statistics. Springer, 2006.

[14] H. Furstenberg and H. Kesten. Products of random matrices. *The Annals of Mathematical Statistics* 31 (1960), 457–469.

[15] P. Bougerol and J. Lacroix. *Products of Random Matrices with Applications to Schrödinger Operators.* Progress in Probability and Statistics. Birkhäuser, 1985.

[16] M. Viana. *Lectures on Lyapunov Exponents.* Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2014.

[17] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, 1995.

[18] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982.

OP code	Open-problem family	Resolved statement in the certified class	Pipeline artifact
OP–JSR–RU	Decidability of robust switching stability / JSR-type questions [1, 3]	Membership in RU-Stable is decidable: DecideRUStable returns either a robust contraction certificate (with ε^* and explicit rate degradation) or an explicit obstruction witness (OG1–OG4), including the avoidance-cycle witness for recurrence failure (OG4).	`robust_certificate` or `obstruction_witness`
OP–MATHER–EFF	Mather-set structure for matrix sequences (ergodic optimization for JSR) [2]	Extremal itineraries admit an explicit finite-state model computed by the pipeline: the boundary carrier $\Sigma_{\mathcal{A}}$ is a computable SFT (Proposition 15.1), and for a candidate core H and word w the recurrence layer returns either a bounded-gap modulus (RC1) or a finite avoidance-cycle witness (OG4), producing a verifiable sofic model (Theorem 15.3).	`MatherSoficModel`
OP–CHAMBER–STAB	Generic finiteness behavior for JSR and chamber regularity [4]	The region-level frontier atlas exports a finite regime stratification together with a single carry-forward decision radius per chamber. In particular, each chamber label is constant on a neighborhood of size at least `eps_dec` around every point of the chamber's certified subcover (Theorem 22.3 and Theorem 35.9). Thus away from margin-collapse loci (notably $\gamma \to 0$, $\delta \to 0$, or support-pattern changes) the tight core, obstruction regime, and contraction rates are locally constant in an explicitly certified sense.	`FrontierAtlasReport`
OP–BOUNDARY–LOC	Devil's-staircase boundary complexity vs. tame regions [4]	Complex behavior is confined to explicit boundary loci where the certificate margins collapse. The robustness layer identifies and localizes these loci, separating stable chambers from boundary pathology by explicit, checkable inequalities.	Region report

Table 1: Open-problem targets supported by the robust certificate program within the certified positive-template class (part 1).

OP code	Open-problem family	Resolved statement in the certified class	Pipeline artifact
OP–SYNC–IND	Černý-type synchronizing bounds for induced automata [5, 6]	For the specific automata generated by the certificate program (Doeblin-word search and avoidance automata), the pipeline provides explicit state counts and either a bounded-gap modulus B (acyclic avoidance) or an explicit cycle witness. When the induced automata satisfy checkable synchronizing hypotheses, known special-class bounds apply as plug-ins for this subclass.	Automaton + B or cycle
OP–PRIM–LEN	Quantitative primitivity / length of a positive (Doeblin) product [7]	The Doeblin-word search is finite and yields either a positive product witness with strict margin delta a finite certificate of failure (OG3) within the declared tight language. When a witness exists, Algorithm 6 returns a shortest Doeblin word and admits an explicit state-size length bound (Proposition 12.1), giving a worst-case search horizon for positivity on the core face block.	Doeblin word + δ
OP–EXTNORM–CONS	Barabanov/extremal norm geometry and constructive computation [8]	A calibrated projective section together with recurrence (RC1) yields an explicit constructive extremal geometry: a computable contraction mechanism and robustness neighborhood for the extremal dynamics. This provides a checkable analogue of an extremal-norm package on the certified class.	Rate pack + radii
OP–BLANG–DEC	Matrix-semigroup decision problems under bounded-language restrictions [9]	The certificate program restricts products to a constrained language (tight-core admissible words plus avoidance constraints). Within that restricted language, the key decision questions needed for stability reduce to finite automata and finite witnesses, yielding an explicit decidability template compatible with bounded-language decidability phenomena.	Language model
OP–FRONTIER–MAP	Mapping undecidable vs. decidable frontiers for matrix products [1, 3]	The robust certificate contract isolates a decidable island and produces explicit exceptional sets where margins vanish. The obstruction taxonomy states exactly which failures prevent certification and returns corresponding finite witnesses.	`FrontierAtlasReport`

Table 2: Open-problem targets supported by the robust certificate program within the certified positive-template class (part 2).

OP code	Open-problem family	Resolved statement in the certified class	Pipeline artifact
OP–ATLAS–FIN	Finite atlas/certificate dichotomy on parameter families	On any compact normalized family Θ within the certified class, region mode terminates with a *single exported frontier atlas report*: either good-global contraction universality is certified (uniform margins and a uniform paired radius), or the report localizes the boundary/degeneracy region by explicit loci and witness references. This turns "cover-or-atlas" into a stable, reusable artifact.	`FrontierAtlasReport`

Table 3: Open-problem target OP–ATLAS–FIN: the region-level frontier atlas artifact as a witness-producing completion of the cover/atlas principle within the certified class.