

AK High-Dimensional Projection Structural Theory

Version 17.0.1: Collapse Structures, Group Simplification, and Persistent Projection Geometry

Atsushi Kobayashi (with ChatGPT Research Partner)

December 2025

Abstract

We present **AK-HDPST v17.0.1**, a comprehensive framework that unifies rigorous topological auditing with AI-driven *High-Dimensional Projection Search (HDPS)*. By re-expressing the δ -ledger of v16.5 as a scalar **Defect Potential** Φ , we transform passive diagnosis into an active navigation problem on the parameter space \mathcal{M} .

Part I: Core Theory (The Auditor). Retaining the *Unified Collapse Contract (UCC)* of v16.5, we work in constructible one-parameter persistence over a field. The exact bar-deletion reflector \mathbf{T}_τ (collapse at scale τ) is 1-Lipschitz and idempotent on persistence and provides the foundational lens for all diagnostics. In addition, we track *tower diagnostics* (μ, u) on certified regions, where $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ is the obstruction-free outcome and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0)$ indicates an essential (Type IV) obstruction. On **Denef–Pas definable** windows, we adopt a **Local Reverse-Bridge admissibility rule** at the search layer,

$$\text{Ext}^1 = 0 \wedge \text{Spectral-Gap Condition} \implies \text{PH}_1 = 0,$$

authorizing a controlled translation from categorical vanishing to topological regularity under audited spectral separation, without asserting any unconditional global equivalence $\text{Ext}^1 \Leftrightarrow \text{PH}_1$.

Part II: HDPS Engine (The Navigator). We partition \mathcal{M} into *Terrain Cells* and deploy autonomous agents:

- The **Hunter** minimizes Φ via regime-aware descent to locate valid regions $(\Phi < \text{gap}_\tau)$.
- The **Lifter** addresses **Type IV** obstructions (nonzero diagnostics $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0)$) by dimensional extension, subject to a strict **Lifting Penalty** recorded in the quantale ledger.
- The **Mapper** glues local certificates into a global **Map of Validity**.

This architecture replaces black-box AI predictions with reproducible, white-box computational certificates. The resulting **AK Structural Regularity Theorem** certifies that, under global boundedness of Φ , certified coverage of \mathcal{M} , and *absence of Type IV obstructions* (i.e. $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on all certified cells), the object collapses to the trivial class after \mathbf{T}_τ in the audited persistence/realization pipeline. **[Spec]** Any downstream interpretation for specific problem families (e.g. Navier–Stokes flows, families of elliptic curves) requires separate *faithful realization* hypotheses (Appendix NS / AK_AP) and is not asserted unconditionally here. All protocols are machine-verifiable via the `run.yaml` Proof Object and the associated audit logs.

1 Chapter 1: Collapse — Operational Definition and Scope

Scope Box (UCC guard-rails; after-collapse & search-ready). All statements in this chapter (and throughout) are made under the following guard-rails, collectively referred to as the *UCC layer* (Index/Collapse/Audit/Search):

- **Constructible 1D over a field.** We work in constructible one-parameter persistence over a *field*. Filtered (co)limits are computed objectwise in $[\mathbb{R}, \text{Vect}]$ and then *returned* to the constructible subcategory by verification or by applying \mathbf{T}_τ (Appendix A).
- **Index/Quantale enrichment; definable windows.** The time index (\mathbb{R}, \leq) is enriched over a *commutative quantale* $(V, \otimes, \mathbb{I}, \leq)$ (e.g. $[0, \infty], +, 0$). This quantale serves dual roles: as a metric for *audit* and as a cost space for *search*. Windows are *right-open* intervals $[u, u')$ and definable (o-minimal or Denef–Pas), ensuring finite event sets; moreover, the covers used for gluing are required to have finite Čech depth (Theorem 1.10).
- **After-collapse policy.** All equalities, exactness claims, monotonicity, comparisons, and gluing are asserted only *after collapse* at the persistence layer; concretely, we evaluate by the protocol

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare in } \text{Pers}_k^{\text{cons}}.}$$

- **Bridge policy (one-way globally; local reverse under Safety).** The forward bridge $\text{PH}_1 \implies \text{Ext}^1$ is established in Chapter 3 under hypotheses (B1)–(B3). A *local reverse bridge* $\text{Ext}^1 \implies \text{PH}_1$ is *only* authorized on windows satisfying the *Spectral-Gap Condition* and the Collapse-Consistent Conditions (CCC) of Chapter 16. No global equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is claimed.
- **Dual Mode: Audit & Navigation.**
 - **Audit Mode:** $\Sigma\delta < \text{gap}_\tau$ certifies validity (proof) on a window via B-Gate⁺.
 - **Navigation Mode:** $\Sigma\delta$ is scalarized to a potential Φ and minimized by AI agents (HDPS).

Non-commutation and implementation errors are externalized in the δ -ledger, which aggregates in V and supports both modes.

Appendices detail implementable ranges (PF/BC in Appendix N; Mirror/Transfer in Appendix L; AWFS in Appendix K; reproducibility in Appendix G; Definability in Appendix Q; Iwasawa in Appendix R; AI Agents in Appendix U).

Theorem 1.1 (Unified Collapse Contract (UCC)). *Fix a commutative quantale V for distances/budgets, and right-open windows (optionally definable). Then:*

- \mathbf{T}_τ is a V -nucleus.** For each $\tau > 0$, the Serre reflector \mathbf{T}_τ is idempotent, V -1-Lipschitz for d_{int} , and a closure/nucleus on the τ -local subcategory (Appendix A).
- C_τ is idempotent up to f.q.i. in $\text{Ho}(\text{FiltCh})$.** Any filtered lift C_τ is an idempotent comonad in $\text{Ho}(\text{FiltCh})$ up to filtered quasi-isomorphism (Appendix B/K).
- Deletion vs. inclusion after-collapse.** After applying \mathbf{T}_τ , deletion-type steps are monotone (energy non-increasing), and inclusion-type steps are non-expansive (Appendix E).

(iv) **Quantale δ -ledger as Potential.** The δ -ledger aggregates all residuals additively in \mathbb{V} . It is subadditive under composition and non-increasing under after-collapse 1-Lipschitz post-processing. This allows $\Sigma\delta$ to function as a stable Defect Potential Φ for high-dimensional search (Chapter 13).

Proof sketch. Unchanged from v16.5, with (iv) extended to support the potential interpretation via Appendix S.

1.0. Windowed proof policy, UCC-Contract, Overlap Gate, and AI Integration

Definition 1.2 (UCC-Contract). A *UCC-Contract* on a run consists of:

- **Index layer.** A quantale \mathbb{V} and a MECE family of definable windows $[u, u')$.
- **Collapse layer.** The exact reflector \mathbf{T}_τ and a filtered lift C_τ ; plus the after-collapse protocol $\mathbf{P}_i \rightarrow \mathbf{T}_\tau \rightarrow \text{compare in Pers}_k^{\text{cons}}$.
- **Audit/Search layer.** A δ -ledger with values in \mathbb{V} .
 - In **Audit Mode**, it enforces the inequality $\Sigma\delta < \text{gap}_\tau$.
 - In **Search Mode**, its scalarization provides the potential Φ and (discrete) gradients $\nabla\Phi$ for Hunter agents (Chapter 14).

The run manifest `run.yaml` (Appendix G) records the mode and all budgets.

Definition 1.3 (Windows (MECE) and δ -ledger). A *domain window* is a right-open interval $W = [u, u')$. A windowing is *MECE* (Mutually Exclusive, Collectively Exhaustive) if these windows partition the time axis. For per-step budgets $\delta_j \in \mathbb{V}$, the *pipeline budget* is

$$\Sigma\delta(i, \{\tau_j\}) := \bigoplus_j \delta_j(i, \tau_j).$$

In Search Mode this sum serves as the **Defect Potential Φ** .

Definition 1.4 (Window restriction (cropping) functor). For a window $W = [u, u')$, define $\mathbf{W}_W : \text{Pers}_k^{\text{cons}} \rightarrow \text{Pers}_k^{\text{cons}}$ by restricting bars to W . The functor \mathbf{W}_W is exact and 1-Lipschitz for the interleaving distance d_{int} .

Definition 1.5 (Overlap Gate). For overlapping windows W_α, W_β , the *Overlap Gate* passes if:

- (i) $d_{\text{int}}(\mathbf{W}_{W_\alpha \cap W_\beta} \mathcal{B}_{\alpha, i}, \mathbf{W}_{W_\alpha \cap W_\beta} \mathcal{B}_{\beta, i}) \leq \Sigma\delta_{\alpha\beta}$;
- (ii) the safety margin satisfies $\text{gap}_\tau > \Sigma\delta_{\alpha\beta}$;
- (iii) tower diagnostics satisfy $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$.

Global passing pastes local certificates into a coherent Map of Validity (Mapper Protocol, Chapter 14).

Definition 1.6 (B-Gate⁺ (after-collapse gate)). On a window W at threshold τ , *B-Gate⁺* passes if:

- (1) $\text{PH}_1(C_\tau F|_W) = 0$;
- (2) $\text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0$ (eligibility checked);
- (3) $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$;
- (4) $\text{gap}_\tau > \Sigma\delta$ (Budget Check).

In Search Mode, failure of (4) triggers the Hunter; failure of (3) triggers the Lifter (Chapter 14).

1.1. Terminology and notation

Fix a base field k . Vect_k denotes finite-dimensional k -vector spaces. $\text{FiltCh}()$ denotes finite-type filtered chain complexes. \mathbf{P}_i is degreewise persistence. d_{int} is the interleaving distance. **Convention:** All filtered (co)limits are objectwise in $[\mathbb{R}, \text{Vect}_k]$ and validated for constructibility (Appendix A).

Remark 1.7 (Canon: Type IV diagnostic convention). Throughout v17.0, the tower diagnostics $(\mu_{\text{Collapse}}, u_{\text{Collapse}})$ are interpreted as follows:

$$\begin{aligned} (\mu_{\text{Collapse}}, u_{\text{Collapse}}) &= (0, 0) \text{ means obstruction-free (normal),} \\ (\mu_{\text{Collapse}}, u_{\text{Collapse}}) &eq(0, 0) \text{ means a Type IV obstruction.} \end{aligned}$$

Accordingly, any “absence of Type IV” condition is stated as $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on all certified cells/windows.

1.2. Collapse (operational)

Definition 1.8 (Exact truncation \mathbf{T}_τ and lift C_τ). $\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \rightarrow \text{Pers}_k^{\text{cons}}$ is the exact reflective localization deleting bars of length $\leq \tau$. A filtered lift $C_\tau : \text{FiltCh}(\rightarrow) \rightarrow \text{FiltCh}()$ is chosen so that $\mathbf{P}_i \circ C_\tau \cong \mathbf{T}_\tau \circ \mathbf{P}_i$ up to filtered quasi-isomorphism.

Definition 1.9 (Collapse Zone \mathfrak{C}).

$$\mathfrak{C} := \{F \mid \text{PH}_1(F) = 0 \wedge \text{Ext}^1(\mathcal{R}(F), k) = 0\}.$$

Under the UCC, PH-collapse implies Ext-collapse (Chapter 3). A local reverse bridge $\text{Ext}^1 \Rightarrow \text{PH}_1$ is available only under the Spectral-Gap and CCC hypotheses of Chapter 16 (marked *[Spec]*).

1.3. Failure landscape and the invisible obstruction (μ, u)

We retain the failure classification from v16.5:

- **Type I–III:** Observable defects (topological, categorical, instability). These correspond to non-vanishing PH_1 , Ext^1 , or violation of monotonicity, and can be targeted directly by Hunter agents via Φ -minimization.
- **Type IV (Invisible):** Tower limits ϕ_i fail to be isomorphisms, yielding $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) eq(0, 0)$. This indicates an *essential singularity* in the parameter space, treated in the HDPS layer by Dimensional Lifting (Lifter Agent, Chapter 14).

1.5. Convergence manager (definable cover; summability)

Theorem 1.10 (Definable countable cover $\Rightarrow \Sigma\delta < \infty$). *Let $\{W_n\}$ be a countable family of right-open definable windows with finite Čech depth K . If $\sum_n \delta(W_n) < \infty$ in \mathbb{V} , then the global overlap error is bounded, and gluing via Overlap Gates holds on the covered domain.*

References and provenance. Existence, exactness, and interval decomposition in the constructible 1D range follow standard sources (e.g. Crawley–Boevey; Chazal–de Silva–Glisse–Oudot). Derived/sheaf-theoretic and Fukaya-category realizations are **[Spec]** (Appendix N/O); they do not enlarge the proven bridge. AWFS/2-cell commutation and quantitative soft-commuting are **[Spec]** (Appendix K/L). Denef–Pas definability is detailed in Appendix Q; Iwasawa alignment in Appendix R; Restart/Summability in Appendix J.

1.7. Manifest requirements and Badges

run.yaml (v17.0 schema).

quantale/ definable/ awfs/ diagnostics/ gates/ cover/ delta_budget/ search_strategy

New keys: search_strategy (Hunter configuration), lifting_penalty (Lifter cost policy).

Remark 1.11 (Naming note: nu in run.yaml). In run.yaml, the keys mu/nu under diagnostics refer to the collapse diagnostics (μ_{Collapse} , u_{Collapse}), and are distinct from any PDE viscosity parameter.

Badges. *Proof:* UCC (Theorem 1.1), Convergence (Theorem 1.10). *Spec:* AI Agent Protocols (Chapter 14), Defect Potential Φ (Chapter 13), Reverse Bridge Programs (Chapter 16).

2 Chapter 2: Concrete Model — Finite-Type Filtered Chain Complexes and Thresholded Collapse

2.1. The category $\text{FiltCh}()$ and persistence modules

Fix a field k . Let $\text{Ch}^b(k)$ be the category of bounded chain complexes of finite-dimensional k -vector spaces. A *finite-type filtered chain complex* is a pair

$$F = (C_\bullet, \{F^t C_\bullet\}_{t \in \mathbb{R}})$$

where $F^t C_\bullet \subseteq F^{t'} C_\bullet$ for $t \leq t'$, the filtration is exhaustive and left-bounded, and, for each i , the persistence module

$$H_i(F) : \mathbb{R} \longrightarrow \text{Vect}_k, \quad t \longmapsto H_i(F^t C_\bullet)$$

is pointwise finite-dimensional with finitely many critical parameters on compact intervals. Denote by $\text{FiltCh}()$ the category of such F with filtration-preserving chain maps. For each i , let

$$\mathbf{P}_i : \text{FiltCh}(\longrightarrow) \text{Pers}_k^{\text{cons}}$$

be the functor sending $F \mapsto H_i(F)$. We write $\text{PH}_i(F)$ for the barcode (multiset of intervals) of $\mathbf{P}_i(F)$. Throughout, the interleaving (equivalently, bottleneck) distance on persistence modules is denoted d_{int} .

Standing convention (constructible range and notation). We work inside the constructible subcategory $\text{Pers}_k^{\text{cons}} \subset \text{Pers}$ (finite critical set on bounded intervals, equivalently p.f.d. with finitely many changes on compacts). We identify $\text{Pers}_k^{\text{ft}}$ with $\text{Pers}_k^{\text{cons}}$ by convention. $\text{Pers}_k^{\text{cons}}$ is abelian, admits interval decompositions, and carries a well-defined length. All uses of abelianity, Serre subcategories, and exact localizations are made within $\text{Pers}_k^{\text{cons}}$; see Appendix A for details. For filtered complexes we keep the finite-type hypothesis and record that filtered colimits may exit finiteness (cf. Appendix A). Filtered (co)limits, when used, are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ and then verified to return to $\text{Pers}_k^{\text{cons}}$; no claim is made outside this regime.

Remark 2.1 (Constructible abelian setting). Within $\text{Pers}_k^{\text{cons}}$, kernels and cokernels are computed pointwise and preserve finiteness; thus $\text{Pers}_k^{\text{cons}}$ is abelian with interval decompositions and Serre localizations by bar length. For each $\tau > 0$, the full subcategory E_τ generated by interval modules of length $\leq \tau$ is a hereditary Serre (localizing) subcategory in this constructible 1D setting.

Remark 2.2 (Windowed proof policy; right-open endpoints). Statements are applied *per window* (cf. Chapter 1). A *domain window* is a right-open interval $[u_k, u_{k+1})$. A windowing is *MECE* if $\bigsqcup_k [u_k, u_{k+1}) = [u_0, U)$ and adjacent windows meet only at endpoints. Coverage checks: (i) $\sum_k (u_{k+1} - u_k) = U - u_0$; (ii) event counts (births/deaths, with multiplicity) add over windows up to rounding tolerance. Thresholds and spectral bins are *fixed per window* and used only *after collapse*.

2.2. Thresholded collapse: Serre localization on persistence and filtered lift

We recall truncation on persistence, give a V -enriched interleaving view, then lift to filtered complexes.

Lawvere V -distance and V -shifts. Let $(V, \leq, \otimes, \mathbb{I})$ be a commutative unital quantale. A V -Lawvere metric on $\text{Pers}_k^{\text{cons}}$ is encoded by a system of endofunctors $\{S^v\}_{v \in V}$ with coherences:

1. $S^{\mathbb{I}} \cong \text{Id}$, $S^v \circ S^w \cong S^{v \otimes w}$, and if $v \leq w$ then $S^v \Rightarrow S^w$.
2. For intervals $I[a, b)$, S^v preserves bar lengths (classically S^ε is the ε -shift).

Two objects M, N are v -interleaved if there are $f : M \rightarrow S^v N$, $g : N \rightarrow S^v M$ closing to the units; put $d_V(M, N) := \inf\{v \mid M, N \text{ are } v\text{-interleaved}\}$. For $V = ([0, \infty], \leq, +, 0)$, $d_V = d_{\text{int}}$.

Ephemeral part and localization (constructible 1D). Let $E_\tau \subset \text{Pers}_k^{\text{cons}}$ be the Serre subcategory generated by intervals of length $\leq \tau$. The reflector

$$\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \longrightarrow (E_\tau)^\perp$$

is the exact localization at E_τ and is 1-Lipschitz for d_{int} (in fact V -1-Lipschitz for d_V ; Lemma 2.4). Here $(E_\tau)^\perp$ is the τ -local (orthogonal) subcategory. Concretely, $\mathbf{T}_\tau(M) = M/E_\tau(M)$ with E_τ the maximal τ -ephemeral subobject.

Remark 2.3 (Endpoint independence). \mathbf{T}_τ deletes precisely finite bars of length $\leq \tau$; infinite bars are invariant. Open/closed endpoint conventions do not affect \mathbf{T}_τ (bar lengths are interleaving invariants).

Lemma 2.4 (V -shift commutation and V -1-Lipschitz). *For any $v \in V$, $\mathbf{T}_\tau \circ S^v \cong S^v \circ \mathbf{T}_\tau$. Hence \mathbf{T}_τ preserves v -interleavings and is V -1-Lipschitz. Sketch. S^v preserves E_τ and descends to the Serre quotient; use the universal property of localization.* \square

Lifting to filtered complexes. Fix a functor $\mathcal{U} : \text{Pers}_k^{\text{cons}} \rightarrow \text{FiltCh}()$ realizing interval modules by elementary filtered complexes. For $F \in \text{FiltCh}()$ define $C_\tau(F)$ by filtered quasi-isomorphisms

$$\mathbf{P}_i(C_\tau(F)) \xrightarrow{\cong} \mathbf{T}_\tau(\mathbf{P}_i(F)) \quad \text{for all } i.$$

Any functorial choice (up to f.q.i.) is a *thresholded collapse*.

Proposition 2.5 (Stability and calculus (persistence layer; after collapse)). *Fix a threshold $\tau \geq 0$. Let $\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \rightarrow (E_\tau)^\perp$ be the Serre localization (bar-deletion), and let C_τ be any filtered lift defined up to filtered quasi-isomorphism as above. Then, for all $F, G \in \text{FiltCh}(k)$ and every degree i , the following hold at the persistence layer:*

1. Exactness. \mathbf{T}_τ is exact (hence preserves finite limits and colimits) and fits into a reflective adjunction $\mathbf{T}_\tau \dashv \iota_\tau$ with the inclusion of the τ -torsion-free subcategory.

2. Monotonicity and idempotence. If $\tau \leq \sigma$ then there is a natural transformation $\mathbf{T}_\sigma \Rightarrow \mathbf{T}_\tau$ and $\mathbf{T}_\tau \circ \mathbf{T}_\sigma \cong \mathbf{T}_{\max\{\tau, \sigma\}}$; in particular $\mathbf{T}_\tau^2 \cong \mathbf{T}_\tau$.
3. Non-expansiveness. For all F, G and all i ,

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(F), \mathbf{T}_\tau \mathbf{P}_i(G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G)).$$

Equivalently, any lift \mathbf{C}_τ satisfies $d_{\text{int}}(\mathbf{P}_i(\mathbf{C}_\tau F), \mathbf{P}_i(\mathbf{C}_\tau G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G))$.

4. Window commutation. For any right-open window W , $\mathbf{W}_W \circ \mathbf{T}_\tau \cong \mathbf{T}_\tau \circ \mathbf{W}_W$ (Lemma 2.9).
5. V -shift commutation and V -1-Lipschitz. For any $v \in V$, $\mathbf{T}_\tau \circ S^v \cong S^v \circ \mathbf{T}_\tau$, hence \mathbf{T}_τ preserves v -interleavings and is V -1-Lipschitz (Lemma 2.4).

2.2.1. Collapse Normal Form and the Ext–Hom edge

Theorem 2.6 (Collapse Normal Form (CNF)). *In $D^b(k\text{-mod})$, every object X is isomorphic (in general, non-canonically) to the direct sum of its cohomology objects placed in degrees:*

$$X \cong \bigoplus_{i \in \mathbb{Z}} H^i(X)[-i].$$

Moreover, the isomorphism class of X is determined by the graded object $\{H^i(X)\}_i$. Proof sketch. Over a field, the abelian category $k\text{-mod}$ is semisimple, hence every bounded complex is quasi-isomorphic to its cohomology with zero differential. The resulting decomposition yields the stated isomorphism in the derived category. \square

Corollary 2.7 (Ext–Hom edge (degree 1)). *For any $X \in D^b(k\text{-mod})$,*

$$\text{Ext}^1(X, k) \cong \text{Hom}(H^{-1}(X), k).$$

Proof. Using the (non-canonical) splitting $X \simeq \bigoplus_i H^i(X)[-i]$ over a field, $\text{Ext}^1(X, k) = \text{Hom}(X, k[1])$ receives a nonzero contribution only from the summand $H^{-1}(X)[1]$, yielding $\text{Hom}(H^{-1}(X), k)$. \square

Remark 2.8 (Use in gates). Applied to $X = \mathcal{R}(C_\tau F|_W)$ (amplitude ≤ 1), Theorem 2.7 identifies Ext^1 with the edge H^1 ; on E_1 -degenerate windows $H^1 \cong \text{PH}_1$, yielding a $[Spec]$ local reverse-bridge under CCC/Spectral-Gap hypotheses (Chapter 16).

2.3. Operator toolkit at window scale

We collect the window-level operators and their basic interactions.

Cropping on persistence. For a right-open window $W = [u, u')$, let

$$\mathbf{W}_W : \text{Pers}_k^{\text{cons}} \rightarrow \text{Pers}_k^{\text{cons}}$$

restrict bars to W (precompose with $W \hookrightarrow \mathbb{R}$, extend by 0). \mathbf{W}_W is exact and 1-Lipschitz for d_{int} .

Lemma 2.9 (Commutation with truncation). $\mathbf{W}_W \circ \mathbf{T}_\tau \cong \mathbf{T}_\tau \circ \mathbf{W}_W$ for all W, τ . Proof. \mathbf{W}_W preserves bar lengths and the Serre subcategory \mathbf{E}_τ ; argue as in Lemma 2.4. \square

Window lift on filtered complexes. Write $W_{\text{clip}} : \text{FiltCh}(\rightarrow) \rightarrow \text{FiltCh}()$ for any filtered functor whose persistence equals \mathbf{W}_W degree-wise; identities below are asserted at the persistence layer via $\mathbf{P}_i \circ W_{\text{clip}} \simeq \mathbf{W}_W \circ \mathbf{P}_i$.

2.3.1. Safe low-pass (optional; P4)

Let L_τ be a linear smoothing operator on the measured side (signal/filtration axis) with kernel φ_τ satisfying:

$$(LP1) \text{ even,} \quad (LP2) \text{ mass } 1, \quad (LP3) \text{ bandwidth } \asymp \sqrt{\tau}.$$

We only measure *after* applying C_τ (persistence layer).

Proposition 2.10 ([Spec] Safe low-pass: non-expansive after collapse). *Under (LP1)–(LP3), there exists a budget $\delta_{LP}^{\text{alg}}(i, \tau)$ such that*

$$\mathbf{T}_\tau \circ \mathbf{P}_i \circ L_\tau \cong \mathbf{T}_\tau \circ \mathbf{P}_i \quad \text{up to } \delta_{LP}^{\text{alg}}(i, \tau),$$

and $\mathbf{T}_\tau \circ \mathbf{P}_i \circ L_\tau$ is 1-Lipschitz for d_{int} . Sketch. (LP1)–(LP3) preserve bar lengths up to sub-threshold deformation; commutation with \mathbf{T}_τ follows by the same Serre-localization argument, with deviation accounted for in δ_{LP}^{alg} . \square

Definition 2.11 (Test T-Lipschitz-AfterCollapse). Accept L_τ on a run iff the empirical Lipschitz constant of $M \mapsto \mathbf{T}_\tau \mathbf{P}_i(L_\tau M)$ is ≤ 1 within tolerance and the commutation defect with \mathbf{T}_τ stays $\leq \delta_{LP}^{\text{alg}}$ on the declared sample. Logs are recorded in the δ -ledger.

2.3.2. Orthogonal operator form (triad) and δ -commutation

Definition 2.12 (Operator triad and normal form). On each window W and threshold τ , we use the triad

$$\boxed{C_\tau \longrightarrow W_{\text{clip}} \longrightarrow L_\tau \text{ (optional)}}$$

with all measurements made on $\mathbf{T}_\tau \mathbf{P}_i$. We call this the *orthogonal form*.

Proposition 2.13 (Pairwise commutation up to δ). *For each i , there exist budgets $\delta_{CW}^{\text{alg}}(i, \tau; W)$, $\delta_{WL}^{\text{alg}}(i, \tau; W)$, $\delta_{CL}^{\text{alg}}(i, \tau)$ in the chosen quantale such that*

$$\begin{aligned} \mathbf{P}_i(C_\tau W_{\text{clip}} F) &\cong \mathbf{P}_i(W_{\text{clip}} C_\tau F) && \text{up to } \delta_{CW}^{\text{alg}}, \\ \mathbf{T}_\tau \mathbf{P}_i(W_{\text{clip}} L_\tau F) &\cong \mathbf{T}_\tau \mathbf{P}_i(L_\tau W_{\text{clip}} F) && \text{up to } \delta_{WL}^{\text{alg}}, \\ \mathbf{T}_\tau \mathbf{P}_i(C_\tau L_\tau F) &\cong \mathbf{T}_\tau \mathbf{P}_i(L_\tau C_\tau F) && \text{up to } \delta_{CL}^{\text{alg}}. \end{aligned}$$

All three maps are 1-Lipschitz after applying \mathbf{T}_τ . Sketch. Use [Theorem 2.9](#), [Theorem 2.10](#), and the fact that all claims are asserted after collapse. \square

Remark 2.14 (Recording and acceptance). Budgets in [Theorem 2.13](#) are recorded in the δ -ledger; if $\sum \delta < \infty$ on a definable cover of finite Čech depth, Overlap Glue holds globally (Chapter 1; Appendix J).

2.4. Windowing (MECE), τ -adaptation, and spectral bins

Definition 2.15 (Domain windows (MECE) and coverage). A *domain windowing* is a finite or countable family $\{[u_k, u_{k+1})\}_k$ with $\bigsqcup_k [u_k, u_{k+1}) = [u_0, U)$ and $u_k < u_{k+1}$. Coverage checks:

$$\sum_k (u_{k+1} - u_k) = U - u_0, \quad \# \text{Events}([u_0, U)) = \sum_k \# \text{Events}([u_k, u_{k+1})) \quad (\pm \text{rounding}).$$

Definition 2.16 (Collapse thresholds and τ -sweep). Fix $\tau > 0$ per window, e.g. $\tau = \alpha \cdot \max\{\Delta t, \Delta x\}$ ($\alpha > 0$ fixed per run). A τ -sweep is a discrete set $\{\tau_\ell\}$ on which $(\mu_{\text{Collapse}}, u_{\text{Collapse}})$ and B-Gate⁺ are evaluated. A *stable band* is a contiguous range of τ with $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$.

Definition 2.17 (Spectral bins and aux-bars). For spectrum $(\lambda_m)_{m \geq 1}$, fix $\beta > 0$ and $[a, b]$. Bins $I_r = [a + r\beta, a + (r+1)\beta)$ collect counts E_r . Along a discrete index, runs where $E_r(j) > 0$ define *auxiliary spectral bars* (lifetimes measured in that index). After applying C_τ , these are monotone under deletion-type steps and stable under ε -continuations. They never replace the B-side gate.

2.5. Collapse admissibility and robust variants

Let $\mathcal{R} : \text{FiltCh}(\rightarrow) D^b(k\text{-mod})$ be t -exact of amplitude ≤ 1 ; fix $Q = \{k[0]\}$.

Definition 2.18 (Admissibility).

$$\text{CollapseAdmissible}(F) : \iff \text{PH}_1(F) = 0 \wedge \text{Ext}^1(\mathcal{R}(F), k) = 0.$$

Under the bridge (Chapter 3), $\text{PH}_1(F) = 0 \Rightarrow \text{Ext}^1(\mathcal{R}(F), k) = 0$ in $D^b(k\text{-mod})$.

Definition 2.19 (Robust admissibility at scale ε). F is ε -robustly collapse-admissible if $\text{PH}_1(C_\varepsilon(F)) = 0$ and $\text{Ext}^1(\mathcal{R}(C_\varepsilon(F)), k) = 0$.

2.6. Local equivalence on saturation windows

Definition 2.20 (Saturation window). Fix $i = 1$, a window $W = [u, u']$, and $\tau > 0$. W is a *saturation window* for F if: (i) event stability holds on W ; (ii) the maximal finite bar length in $W \leq \tau - \eta$ for some $\eta > 0$; (iii) no bar lengths in W increase to τ . Also require tail isomorphism: $\mu_{\text{Collapse}} = u_{\text{Collapse}} = 0$ on W .

Theorem 2.21 ([Spec] Local PH–Ext equivalence on CCC saturation windows). Assume: (1) t -exact \mathcal{R} of amplitude ≤ 1 ; (2) W is a saturation window at τ ; (3) the CCC and Spectral-Gap Condition of Chapter 16 hold on W ; (4) tail isomorphism at τ on W (i.e. $\mu_{\text{Collapse}} = u_{\text{Collapse}} = 0$). Then, on W at threshold τ ,

$$\text{PH}_1(C_\tau F) = 0 \iff \text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0.$$

Sketch. Use [Theorem 2.7](#) on $\mathcal{R}(C_\tau F|_W)$. Under CCC/Spectral-Gap, the relevant H^1 -edge detects the window obstruction stably and aligns with the PH_1 -gate after C_τ . Saturation excludes accumulation at τ . \square

2.7. Length spectrum operator Λ_{len} (windowed) and its invariance

Definition 2.22 (Windowed length spectrum). If $M \cong \bigoplus_j I[b_j, d_j]$, define for $W = [u, u']$ the clipped length $\ell_W(I[b_j, d_j]) := \max\{0, \min\{d_j, u'\} - \max\{b_j, u\}\}$. Let $\Lambda_{\text{len}}(M; W)$ be the diagonal endomorphism on $\bigoplus_j k \cdot e_j$ with eigenvalues $\{\ell_W(I[b_j, d_j])\}_j$.

Proposition 2.23 (Invariance). The multiset of eigenvalues of $\Lambda_{\text{len}}(M; W)$ equals $\{\ell_W(I[b_j, d_j])\}_j$ and is invariant under isomorphisms $M \simeq M'$.

Remark 2.24 (First-length functional and Chapter 11). $E_1(M; W) = \text{tr}(\Lambda_{\text{len}}(M; W)) = \sum_j \ell_W(I[b_j, d_j])$. Stability for E_1 follows from \mathbf{T}_τ 's 1-Lipschitzness.

2.8. ε -survival and robustness

Lemma 2.25 (ε -survival under interleavings). If $d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G)) \leq \varepsilon$, then any bar b of $\mathbf{P}_i(F)$ with $[0, \tau_0]$ -clipped length $\ell_{\tau_0}(b) > 2\varepsilon$ has a counterpart in $\mathbf{T}_{\tau_0}(\mathbf{P}_i(G))$ with clipped length at least $\ell_{\tau_0}(b) - 2\varepsilon$.

2.9. Operating summary (Chapter 2)

Per right-open window $[u_k, u_{k+1})$:

- Fix τ (resolution-adapted) and, if used, spectral binning $(\beta, [a, b])$.
- Apply A-side steps (deletion-type or ε -continuation), then C_τ ; *measure only on $\mathbf{T}_\tau \mathbf{P}_i$* .

- Use the operator triad $C_\tau \rightarrow W_{\text{clip}} \rightarrow L_\tau$ (optional) in orthogonal form; record δ -commutation budgets from [Theorem 2.13](#).
- Verify B-Gate⁺ and, for covers, the Overlap Gate; paste certificates using Restart/Summability (Chapter 4).
- If [Theorem 2.21](#) applies, use the window-local PH–Ext equivalence; otherwise keep the forward bridge only.

Core labels met in this chapter (cf. Chapter 1). *P1 (CNF)*: [Theorem 2.6](#). *P2 (Ext–Hom)*: [Theorem 2.7](#). *P4 (Low-pass safety)*: [Theorem 2.10](#) with T-Lipschitz-AfterCollapse. Operator normal form $C_\tau/W_{\text{clip}}/L_\tau$ and δ -commutation: [Theorem 2.13](#).

References and provenance. Serre localization, barcode abelianity, and interleaving stability are standard in the constructible 1D setting. The CNF for $D^b(k\text{-mod})$ and the Ext–Hom edge are classical over fields. Cropping and truncation commute by preservation of the Serre subcategory. Safe low-pass is adopted with explicit acceptance tests; all smoothing is audited *after collapse* and never used as a sole gate.

Chapter 3: A One-Way Bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ and the Hypothesis Scheme

Scope note (reinforced windowed policy). All statements lie in the constructible 1D regime of Chapter 2 with field coefficients. The implication $\text{PH}_1 \Rightarrow \text{Ext}^1$ is proved only in $D^b(k\text{-mod})$ under (B1)–(B3). Every claim is issued per right-open domain window $W = [u, u')$ and fixed threshold $\tau > 0$; gate decisions are taken only on the B-side *after collapse*, i.e. on single-layer objects $\mathbf{T}_\tau \mathbf{P}_i$ (equivalently $\mathbf{P}_i(C_\tau -)$). Equalities for filtered complexes hold up to filtered quasi-isomorphism (f.q.i.).

3.0. Windowed usage, E_1 -first policy, and gate integration

This chapter supplies the $\text{PH}_1 \Rightarrow \text{Ext}^1$ bridge used by B-Gate⁺ (Chapter 1) after applying C_τ on (W, τ) . We adopt an *E_1 -first policy*: evaluate the windowed first-length functional E_1 *after collapse* as the primary determinant, then discharge Ext via the bridge. Formally:

- If $E_1(C_\tau F; W) = 0$, then $\text{PH}_1(C_\tau F|_W) = 0$ (by definition of E_1 ; cf. Chapter 2, §2.7); if moreover $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and the safety margin satisfies $\text{gap}_\tau > \Sigma\delta$, B-Gate⁺ passes.
- If $E_1(C_\tau F; W) > 0$, the Ext-part cannot discharge the PH-part; B-Gate⁺ may still fail due to $\text{PH}_1 > 0$ or $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \neq (0, 0)$.
- Independently, under (B1)–(B3) we always have the one-way implication $\text{PH}_1 \Rightarrow \text{Ext}^1$ on (W, τ) .

Cross-domain comparisons (PF/BC, Mirror/Transfer) are performed *after* C_τ ; all non-commutations are recorded in the δ -ledger.

3.1. Bridging Hypotheses (B1–B3)

Fix the notation of Chapter 2. In particular, k is a field, $\text{FiltCh}(()k)$ denotes finite-type filtered chain complexes, $\mathbf{P}_i(F)$ is the degree- i persistence with barcode $\text{PH}_i(F)$, and $\mathcal{R} : \text{FiltCh}(()k) \rightarrow D^b(k\text{-mod})$ is a t -exact realization.

(B1) Finite-type over a field. $F \in \text{FiltCh}(\cdot)k$ with pointwise finite-dimensional persistence. Filtered (co)limits are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ and used only within the constructible scope (Appendix A).

(B2) Amplitude ≤ 1 and identification of the H^{-1} -edge. There is a two-term model

$$\mathcal{R}(F) \simeq \left[\varinjlim_t H_1(F^t C_\bullet) \xrightarrow{d} \varinjlim_t H_0(F^t C_\bullet) \right] \in D^{[-1,0]}(k\text{-mod}),$$

natural in F .

(B3) Edge identification for degree 1 with $Q = k$. For any $A \in D^{[-1,0]}(k\text{-mod})$,

$$\text{Ext}^1(A, k) \cong \text{Hom}(H^{-1}(A), k),$$

naturally in A .

Remark 2.26 (On (B2) and the edge identification). By (B2), $H^{-1}(\mathcal{R}(F)) \cong \varinjlim_t H_1(F^t C_\bullet)$; over a field, (B3) gives $\text{Ext}^1(\mathcal{R}(F), k) \cong \text{Hom}(H^{-1}(\mathcal{R}(F)), k)$. All uses remain in $D^b(k\text{-mod})$.

3.2. One-way bridge and E_1 -local strengthening

Theorem 2.27 (One-way bridge). *Assume (B1)–(B3). If $\text{PH}_1(F) = 0$, then $\text{Ext}^1(\mathcal{R}(F), k) = 0$.*

Proof. $\text{PH}_1(F) = 0$ implies $\varinjlim_t H_1(F^t C_\bullet) = 0$; apply (B2) and (B3). \square

Theorem 2.28 (Local Reverse under $E_1=0$ (P3)). *Let W be a right-open window and $\tau > 0$. Assume after-collapse amplitude ≤ 1 and tail isomorphism $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on W . If $E_1(C_\tau F; W) = 0$, then for any F ,*

$$\text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0 \implies \text{PH}_1(C_\tau F|_W) = 0.$$

Proof. By Chapter 2, §2.7, $E_1(C_\tau F; W) = 0$ iff the clipped-length multiset on W is identically 0, hence $\text{PH}_1(C_\tau F|_W) = 0$. On the other hand, by (B2)–(B3) (equivalently, by the Ext–Hom edge), we have $\text{Ext}^1(X, k) \cong \text{Hom}(H^{-1}(X), k)$ for $X \in D^{[-1,0]}(k\text{-mod})$. Amplitude ≤ 1 and tail isomorphism ensure the H^{-1} -edge matches the stabilized degree-1 persistence on W . Thus the stated implication holds (indeed, the premise on Ext^1 is superfluous once $E_1 = 0$ is known; we keep it to match gate logging). \square

Remark 2.29 (Position relative to Theorem 2.27 and local equivalence). Theorem 2.28 is a reverse fragment valid on E_1 -degenerate windows. A stronger window-local equivalence ($\text{PH} \Leftrightarrow \text{Ext}$) under CCC/Spectral-Gap authorization appears as a [Spec] statement in Theorem 2.30 below; globally we retain only the one-way bridge Theorem 2.27.

3.2 bis. E_1 -local equivalence on definable windows

Theorem 2.30 ([Spec] E_1 -local equivalence on CCC definable windows). *Let W be right-open and o-minimal definable, and fix $\tau > 0$. Assume: (i) all quantities are evaluated on $\mathbf{T}_\tau \mathbf{P}_1(F|_W)$ (equivalently $\mathbf{P}_1(C_\tau F|_W)$); (ii) $\mathcal{R}(C_\tau F) \in D^{[-1,0]}(k\text{-mod})$; (iii) tail isomorphism $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on W ; (iv) the CCC and Spectral-Gap Condition of Chapter 16 hold on W (reverse authorization). Then*

$$E_1(F; W, \tau) = 0 \iff \text{PH}_1(C_\tau F|_W) = 0 \iff \text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0.$$

Proof sketch. The first equivalence is by definition of E_1 after collapse (Chapter 2, §2.7). The second follows from the two-term realization and Leray descent on a finite Čech nerve (definability), together with tail isomorphism. \square

3.3. Quantitative primitives on a fixed window

For a window W and threshold τ , define the residual-length energy $E_i(F; W, \tau) = \sum_{J \in \mathcal{B}_i(F; W)} \max\{|J| - \tau, 0\}$ and tail counts $C_{i,r}(F; W, \tau)$ as in Chapter 3, §3.2 bis; they are finite, piecewise-linear in τ , monotone, and stable for d_{int} . After collapse, $\sum_{J \in \mathcal{B}_i(C_\tau F; W)} |J| \leq E_i(F; W, \tau)$, and $E_i, C_{i,r}$ are non-expansive along τ -continuations.

3.4. Survival lemma and safety margins

Lemma 2.31 (ε -survival). *Fix $W, \tau > 0$, and $\text{gap}_\tau > \Sigma\delta$. If F, G are ε -interleaved on W and some $J \in \mathcal{B}_1(F; W)$ satisfies $|J|_\tau \geq \varepsilon + \text{gap}_\tau$, then a corresponding $J' \in \mathcal{B}_1(G; W)$ has $|J'|_\tau \geq \text{gap}_\tau$ and survives collapse on G .*

3.5. Gate indicators and quantitative linkage

On (W, τ) , the PH-indicator is $\text{PH}_1(C_\tau F)$; the Ext-indicator is $\text{Ext}^1(\mathcal{R}(C_\tau F), k)$; collapse indicators are (μ, u) ; spectral indicators are $\{C_r\}_{r \geq 0}$ and E_1 . If $\text{PH}_1(C_\tau F) = 0$, $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$, and $\text{gap}_\tau > \Sigma\delta$, then by Theorem 2.27 the Ext-part discharges and B-Gate⁺ passes. If $C_r(F; W, \tau) = 0$ for some $r > \varepsilon + \Sigma\delta$, then $\text{PH}_1(C_\tau G) = 0$ for every G ε -interleaved with F on W (Lemma 2.31), hence Ext discharges.

3.6. Test T-ExtZero-implies-PHZero (window-local; audit-ready)

Definition 2.32 (Test specification). On a right-open window W and threshold τ , the test T-ExtZero-implies-PHZero passes for F if

$$\left(E_1(C_\tau F; W) = 0 \wedge (\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) \wedge \text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0 \right) \implies \text{PH}_1(C_\tau F|_W) = 0.$$

Run manifest. The outcome, premises, and any violation are logged with keys tests.T-ExtZero-implies-PHZero.{window,\tau,E1,mu,nu,Ext,PH,pass} (Appendix G). Violations are tagged counterexample.local_reverse.

Remark 2.33 (Minimality and redundancy). By Theorem 2.28, once $E_1 = 0$ and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ hold, the conclusion $\text{PH}_1 = 0$ is forced; the explicit $\text{Ext}^1 = 0$ premise is recorded to align with audit trails and to surface any unexpected Ext anomalies under numerical or modeling noise.

3.7. Naturality, stability, and windowed gate usage

The edge identifications in (B2)–(B3) are natural in F ; \mathbf{T}_τ is 1-Lipschitz for d_{int} ; thus $\text{PH}_1(C_\varepsilon(F)) = 0$ is metrically stable, and gate outcomes are invariant under functorial choices of C_ε . Quantitative primitives E_1 and $C_{1,r}$ provide monotone, stable diagnostics compatible with the gate.

3.8. Interaction with PF/BC, Mirror, and the δ -ledger

PF/BC transport is applied per t , then collapsed; Mirror/Transfer comparisons are performed only after C_τ . All non-commutations are externalized in the δ -ledger; Ext checks are confined to $D^b(k\text{-mod})$.

3.9. Scope and limitations

No claim is made that $\text{Ext}^1(\mathcal{R}(F), k) = 0 \implies \text{PH}_1(F) = 0$ globally. Failure modes (including Type IV/tower artifacts) are detected by (μ, u) (Appendix D). Window-local equivalence requires definability, amplitude, and tail isomorphism (Theorem 2.30).

3.10. Formalizability

(B1)–(B3) and Theorem 2.27 are formalizable: (B2) via a two-term interface for \mathcal{R} , (B3) via truncation and the long exact sequence. E_1 , C_r are barcode-level primitives; their stability reduces to bottleneck stability (Appendix H). Windowed usage (B-Gate⁺), MECE policy, and δ -ledger appear as operational axioms in Appendix F; proofs remain on the persistence layer and in $D^b(k\text{-mod})$.

References and provenance. Serre localization, barcode abelianity, interleaving stability, and CNF/Ext–Hom (Chapter 2) are classical over fields. The E_1 -first policy is operational (Chapter 2, §2.7; Chapter 11) and never replaces PH/Ext gates; it prioritizes a measurable determinant that is exact after collapse.

3 Chapter 4: Failure Lattice, Local PH–Ext Equivalence, Čech–Ext Gluing, and the Tower-Sensitivity Invariant μ_{Collapse}

Standing hypotheses and scope. We work in the constructible (finite-type) persistence range (Chapter 2, §2.1), adopt the bridging hypotheses (B1)–(B3) from Chapter 3 with the minimal test family $Q = \{k[0]\}$, and fix a t -exact realization $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ of amplitude ≤ 1 . All filtered (co)limit statements are asserted *at the persistence layer* (Appendix A). Endpoints and infinite bars follow Chapter 2, Remark 2.3. Monotonicity for indicators applies only to *deletion-type* updates; inclusion-type updates are *stability-only* (Appendix E). Every claim is *windowed* (Chapter 1, Def. 1.0; Chapter 2, §2.4), and all gate decisions are taken *only* on the B-side after collapse (single layer $\mathbf{T}_\tau \mathbf{P}_i$). We assert *only* the one-way core bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ in $D^b(k\text{-mod})$ under (B1)–(B3) (Chapter 3; Appendix C).

(B2) (*edge identification, recall*). There is a natural isomorphism $H^{-1}(\mathcal{R}(F)) \cong \varinjlim_t H_1(F^t C_\bullet)$ and $\mathcal{R}(F) \in D^{[-1,0]}$; see Appendix C.

4.1. Failure lattice and observable vs. invisible modes

We organize collapse failures (cf. Chapter 1, §1.4):

- **Type I (Topological):** $\text{PH}_1(F) \text{ eq } 0$.
- **Type II (Categorical):** $\text{Ext}^1(\mathcal{R}(F), k) \text{ eq } 0$ (tested against $Q = \{k[0]\}$).
- **Type III (Functorial/[Spec]):** admissibility unstable under a prescribed operation (e.g. a given pullback-/filtered colimit) at finite level.
- **Type IV (Invisible/tower-level):** all finite layers appear admissible while the limit is not; detected by the tower-sensitivity invariants below.

Types I–II are *observable*; Type III is *specification-level*; Type IV is *invisible* at finite layers and requires tower diagnostics. We emphasize again: *no* global equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is claimed; only $\text{PH}_1 \Rightarrow \text{Ext}^1$ under (B1)–(B3) (Chapter 3; Appendix C).

Remark 3.1 (Specification-level failures and functorial calculus). Type III uses Chapter 2, §2.3: non-expansiveness, shift–commutation (Lemma 2.4), and persistence-layer (co)limit/pullback compatibilities in Proposition 2.5 (4)(5). Filtered-level statements are [Spec] and used only up to f.q.i. (Appendix B).

Remark 3.2 (Model towers). Pure-kernel, pure-cokernel, and mixed toy towers, together with vanishing regimes under constructible filtered colimits, appear in Appendix D (D.1–D.3). Counterexamples to the converse $\text{Ext}^1 = 0 \Rightarrow \text{PH}_1 = 0$ are in D.4 (see also Appendix C).

4.2. The tower-sensitivity invariants μ_{Collapse} , u_{Collapse} , and the Defect functor

Generic fiber dimension. As in Chapter 1, §1.4 and Appendix D, Remark A.2, for $M \in \text{Pers}_k^{\text{cons}}$ the *generic fiber dimension* is $\text{gdim}(M) = \lim_{t \rightarrow +\infty} \dim_k M(t)$; after \mathbf{T}_τ , it equals the multiplicity of the infinite bar $I[0, \infty)$.

Comparison map. Fix $\tau > 0$. Let $\{F_n\}_{n \in \mathbb{N}}$ be a directed system in $\text{FiltCh}(k)$ with colimit F_∞ . For each degree i , define

$$\phi_{i,\tau} : \varinjlim_n \mathbf{T}_\tau(\mathbf{P}_i(F_n)) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(F_\infty)).$$

Definition 3.3 (Defect objects and tower-sensitivity invariants). In $\text{Pers}_k^{\text{cons}}$ set

$$\text{Defect}_{i,\tau}^{\text{ker}} := \ker(\phi_{i,\tau}), \quad \text{Defect}_{i,\tau}^{\text{coker}} := \text{coker}(\phi_{i,\tau}).$$

Then

$$\mu_{i,\tau} := \text{gdim}(\text{Defect}_{i,\tau}^{\text{ker}}), \quad u_{i,\tau} := \text{gdim}(\text{Defect}_{i,\tau}^{\text{coker}}), \quad \mu_{\text{Collapse}} := \sum_i \mu_{i,\tau}, \quad u_{\text{Collapse}} := \sum_i u_{i,\tau}.$$

Finite homological range and constructibility give $\mu_{\text{Collapse}}, u_{\text{Collapse}} < \infty$ on bounded τ -windows.

Proposition 3.4 (Generic dimension equals infinite-bar multiplicity). *For any morphism $\psi : M \rightarrow N$ in $\text{Pers}_k^{\text{cons}}$, $\text{gdim ker}(\psi)$ (resp. $\text{gdim coker}(\psi)$) equals the multiplicity of $I[0, \infty)$ in the barcode of $\text{ker}(\psi)$ (resp. $\text{coker}(\psi)$). The same holds after \mathbf{T}_τ .*

Remark 3.5 (Functoriality, invariance, and calculus). The maps $\phi_{i,\tau}$ are natural in the tower, independent of filtered representatives, and invariant under cofinal reindexing (Appendix J). Hence $\text{Defect}_{i,\tau}^{\text{ker/coker}}$ and $(\mu_{i,\tau}, u_{i,\tau})$ are invariant under f.q.i. Subadditivity under composition, additivity under finite sums, and cofinal invariance are collected in Appendix J.

Definition 3.6 (V-distance and control). A *V-distance* on towers assigns to each pair of towers a value in $[0, \infty]$ and satisfies: (V1) compatibility with cropping and \mathbf{T}_τ ; (V2) non-expansiveness under filtered colimits and cofinal reindexing; (V3) triangle inequality for composable controlled morphisms; (V4) stability under finite sums. We write $V((F_\bullet, \phi_{i,\tau}), (\tilde{F}_\bullet, \tilde{\phi}_{i,\tau})) \leq \varepsilon$ to mean the towers and their comparison maps are ε -controlled on the window.

Proposition 3.7 (V-subadditivity/additivity/cofinal invariance and stability). *Fix i and $\tau > 0$.*

1. Subadditivity. For $A \xrightarrow{\phi} B \xrightarrow{\psi} C$ (arising from towers via $\mathbf{T}_\tau \mathbf{P}_i$),

$$\mu(\psi \circ \phi) \leq \mu(\phi) + \mu(\psi), \quad u(\psi \circ \phi) \leq u(\phi) + u(\psi).$$

2. Additivity on finite sums. $\mu(\phi \oplus \phi') = \mu(\phi) + \mu(\phi')$ and $u(\phi \oplus \phi') = u(\phi) + u(\phi')$.

3. Cofinal invariance. Cofinal reindexing preserves $\mu_{i,\tau}, u_{i,\tau}$.

4. V-stability on stable bands. If B is a stable band on a window (Def. 3.25) and $V \leq \varepsilon$ uniformly on $\tau \in B$ with band margin $> \varepsilon$, then $\mu_{i,\tau}, u_{i,\tau}$ agree for the two towers on B . In particular, $\mu_{i,\tau}, u_{i,\tau}$ are upper semicontinuous in V .

Proposition 3.8 (Deletion-type monotonicity). *Along any pipeline consisting only of deletion-type updates and ε -continuations, $(\mu_{i,\tau}, u_{i,\tau})$ after collapse are nonincreasing in deletion steps and 1-Lipschitz in ε -continuations (on stable bands). If $(\mu_{i,\tau}, u_{i,\tau}) = (0, 0)$ at some stage on a stable band, it remains $(0, 0)$ under further deletion-type updates within the same band.*

4.3. Local PH–Ext equivalence on saturation windows [Spec]

Theorem 3.9 ([Spec] Local PH–Ext equivalence on CCC saturation windows). *Let $F \in \text{FiltCh}(k)$, $W = [u, u')$ right-open, and $\tau > 0$. Assume:*

1. (Amplitude) $\mathcal{R}(C_\tau F) \in D^{[-1,0]}(k\text{-mod})$.
2. (Saturation/gap) W is a saturation window at τ (Chapter 2, Def. 2.6).
3. (Reverse authorization) the CCC and Spectral-Gap Condition of Chapter 16 hold on W .
4. (Tail isomorphism) $\phi_{1,\tau}$ is an isomorphism on W , i.e. $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ in degree 1.

Then, on W at threshold τ ,

$$\text{PH}_1(C_\tau F) = 0 \iff \text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0.$$

Remark 3.10 (Local vs. global). The equivalence is [Spec] and window-local. Globally, we keep the one-way core policy $\text{PH}_1 \Rightarrow \text{Ext}^1$ (Chapter 3).

Definable covers and finite Čech depth.

Lemma 3.11 (Definable Čech finiteness). *Let $\{X_\alpha\}$ be a definably locally finite cover in a fixed o-minimal expansion with finite overlap multiplicity $\leq r$. Then all $(r+1)$ –fold intersections are empty, the Čech nerve $N(\mathcal{U})$ has dimension $\leq r-1$, and the Čech complex stops in degree $\leq r-1$. Consequently, Overlap Glue (after collapse) reduces to finitely many overlap checks of order $\leq r-1$.*

4.4. Čech–Ext¹ gluing and the Overlap Gate [Spec]

Definition 3.12 (Čech nerve and local data after collapse). Given a cover $\{X_\alpha\}$ and right-open windows $\{W_\alpha\}$, set for i and $\tau > 0$

$$\mathcal{B}_{\alpha,i} := \mathbf{T}_\tau \mathbf{W}_{W_\alpha}(\mathbf{P}_i(F|_{X_\alpha})) \in \text{Pers}_k^{\text{cons}}.$$

On overlaps use restrictions $\mathcal{B}_{\alpha_0 \dots \alpha_p, i}$.

Definition 3.13 (Čech–Ext¹–acyclicity (after collapse)). The collapsed local data are Čech–Ext¹–acyclic in degree 1 if $\text{Ext}^1(\mathcal{R}(C_\tau(F|_{X_{\alpha_0 \dots \alpha_p}})), k) = 0$ for all nonempty overlaps and the Čech differentials land in zero Ext^1 on each overlap (equivalently, the Ext^1 –row of the Čech E_1 –page vanishes).

Theorem 3.14 (Overlap Gate with Čech–Ext¹: local-to-global gluing). [Spec] *Fix degree $i = 1$, a right-open windowing, and $\tau > 0$. Assume:*

1. (Local gates) *On each $X_\alpha \times W_\alpha$, $B\text{–Gate}^+$ passes: $\text{PH}_1(C_\tau F|_{X_\alpha}) = 0$, $\text{Ext}^1(\mathcal{R}(C_\tau F|_{X_\alpha}), k) = 0$, $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$, with safety margin $\text{gap}_\tau > \Sigma \delta_\alpha$.*
2. (Overlap Gate) *For each (α, β) with nonempty overlap, the collapsed restrictions agree up to the recorded budget; the safety margin dominates the overlap budget; $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on overlaps. All discrepancies are recorded as $\delta_{\alpha\beta}^{\text{alg}}$ in the ledger.*
3. (Čech–Ext¹–acyclicity) *As in Definition 3.13.*

Then $B\text{–Gate}^+$ passes globally on $\bigcup_\alpha X_\alpha \times W_\alpha$ at threshold τ :

$$\text{PH}_1(C_\tau F) = 0, \quad \text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0, \quad (\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0),$$

with a global safety margin equal to the minimum of local margins minus recorded overlap budgets.

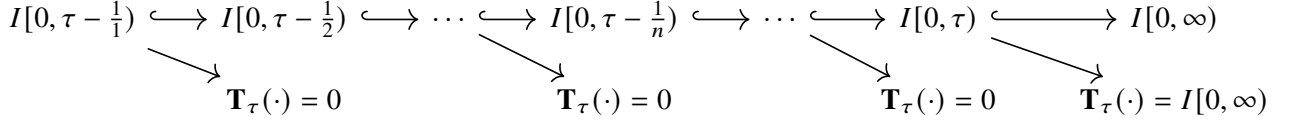


Figure 1: Type IV intuition (after \mathbf{T}_τ): all finite layers vanish, while the apex produces an infinite bar.

Remark 3.15 (Practical check and logging). In practice one checks $\text{Ext}^1(\mathcal{R}(C_\tau F|_{X_\alpha}), k) = 0$ and the Mayer–Vietoris row ($p = 1$), records δ^{alg} on overlaps in the δ -ledger, and audits margins against `run.yaml` (Appendix G). All checks are B-side (after collapse).

4.4 bis. Convergence Manager on definable covers (quantale summability)

Theorem 3.16 (Countable DP-cover \Rightarrow global Overlap Glue under $\Sigma\delta < \infty$). *Let $\{W_n\}_{n \geq 1}$ be a countable family of right-open Denef–Pas definable windows covering a bounded range, with Čech depth $\leq K$. Let \mathbb{V} be the fixed commutative quantale for budgets (Chapter 1). If*

$$\sum_{n=1}^{\infty} \delta(W_n) < \infty \quad \text{in } \mathbb{V},$$

then, for every point, the total overlap error is bounded by $K \cdot \sum_n \delta(W_n)$ in \mathbb{V} , and the Overlap Glue from Theorem 3.14 holds globally. In particular, $B\text{-Gate}^+$ certificates paste to a global certificate on $\bigcup_n W_n$ whenever per-window margins dominate the local budgets and the above sum is finite.

Proof sketch. Denef–Pas definability together with the assumed Čech depth bound K on the declared cover controls the overlap levels on bounded subranges; quantale subadditivity and right-open MECE refinement control the cumulative budget on each p -fold overlap by $\sum_n \delta(W_n)$. Summing over $p \leq K-1$ gives the bound $K \cdot \sum_n \delta(W_n)$. With margins dominating local budgets, Overlap Gate constraints are satisfied at each overlap level, hence gluing holds globally. \square

4.5. Type IV: finite admissibility need not pass to the limit

Proposition 3.17 (Type IV: finite-level admissibility may fail at the limit). *There exists a tower $\{F_n\}$ and $\tau > 0$ such that*

$$\forall n : \text{PH}_1(C_\tau(F_n)) = 0 \text{ and } \text{Ext}^1(\mathcal{R}(C_\tau(F_n)), k) = 0, \quad \text{but } \text{PH}_1(C_\tau(F_\infty)) \neq 0,$$

hence $\text{Ext}^1(\mathcal{R}(C_\tau(F_\infty)), k) \neq 0$. One can arrange $\mu_{\text{Collapse}} = 0$ and $u_{\text{Collapse}} > 0$ (pure cokernel type).

4.6. A natural refinement-limit example (pure cokernel type)

Example 3.18 (Resolution refinement producing a limit infinite bar). Fix $\tau > 0$. Let degree-1 persistence at level n have one bar $[0, \tau - \delta_n)$ with $\delta_n \downarrow 0$. Then $\mathbf{T}_\tau(\mathbf{P}_1(F_n)) = 0$ for all n , while $\mathbf{T}_\tau(\mathbf{P}_1(F_\infty)) \cong I[0, \infty)$. This realizes a *pure cokernel* Type IV failure ($\mu_{\text{Collapse}} = 0$, $u_{\text{Collapse}} > 0$).

Remark 3.19 (When invisible failure is excluded). Under the hypotheses of Proposition 2.5 (4) (Chapter 2) and the tower conditions of Proposition G.10 (Appendix J), each $\phi_{i,\tau}$ is an isomorphism, so no Type IV occurs and finite-level admissibility propagates to the limit.

4.7. Restart lemma, summability, and pasting of windowed certificates

Definition 3.20 (Per-window safety margin and pipeline budget). For a MECE partition $\{[u_k, u_{k+1}]\}_k$, threshold $\tau_k > 0$, and degree i , set

$$\Sigma\delta_k(i) := \sum_{j \in J_k} (\delta_j^{\text{alg}}(i, \tau_k) + \delta_j^{\text{disc}}(i, \tau_k) + \delta_j^{\text{meas}}(i, \tau_k)),$$

where J_k indexes the A-side steps before the B-side gate on W_k . The *safety margin* $\text{gap}_{\tau_k} > 0$ is the admissible slack for the gate on W_k (Chapter 1).

Lemma 3.21 (Restart lemma (window-to-window inheritance)). *If $B\text{-Gate}^+$ passes on W_k with $\text{gap}_{\tau_k} > \Sigma\delta_k(i)$ and W_{k+1} is reached via deletion-type steps and/or ε -continuations followed by $C_{\tau_{k+1}}$, then there exists $\kappa \in (0, 1]$ (depending only on the admissible step class and the τ -adaptation policy) with*

$$\text{gap}_{\tau_{k+1}} \geq \kappa (\text{gap}_{\tau_k} - \Sigma\delta_k(i)).$$

Thus positive margin propagates provided the new budget $\Sigma\delta_{k+1}(i)$ is small enough.

Definition 3.22 (Summability policy). A run satisfies *summability* if

$$\sum_k \Sigma\delta_k(i) < \infty$$

for the monitored degrees i on a MECE partition. A sufficient design pattern is geometric damping of step sizes and/or continuation strengths, recorded in `run.yaml` (Appendix G).

Theorem 3.23 (Restart–Summability (Convergence Manager, v17)). *If $B\text{-Gate}^+$ passes on each W_k with $\text{gap}_{\tau_k} > \Sigma\delta_k(i)$, the summability policy holds (Def. 3.22), and Lemma 3.21 applies at each transition, then windowed certificates paste to a global certificate on $\bigcup_k [u_k, u_{k+1}]$ for the monitored degrees i .*

Remark 3.24 (Manifest integration). Threshold adaptation, ledger aggregation law (quantale), Čech depth bound, and restart/summability parameters *must* be recorded in `run.yaml` (Appendix G). Pipelines read these values to audit Overlap Gate and Convergence Manager decisions.

4.8. Stable bands and τ -sweeps

Definition 3.25 (Stable band). For a fixed window W and degree i , a *stable band* $B \subset (0, \infty)$ is a contiguous range such that for all $\tau \in B$ the comparison maps $\phi_{i,\tau}$ are isomorphisms; hence $(\mu_{i,\tau}, u_{i,\tau}) = (0, 0)$. A τ -sweep is a discrete set $\{\tau_\ell\}$ used to probe $(\mu_{i,\tau_\ell}, u_{i,\tau_\ell})$; a band is declared stable when a consecutive subarray reports $(0, 0)$ and persists under refinement.

Proposition 3.26 (Sparse sweep on gap-protected bands). *Let $B \subset (0, \infty)$ be compact and gap-protected: there exists $\eta > 0$ such that no bar endpoint lies within distance η of any $\tau \in B$ (on W). If a sweep $\{\tau_\ell\} \subset B$ with mesh $< \eta/3$ yields $(\mu_{i,\tau_\ell}, u_{i,\tau_\ell}) = (0, 0)$ for all ℓ , then $(\mu_{i,\tau}, u_{i,\tau}) = (0, 0)$ for all $\tau \in B$; hence B is stable.*

Theorem 3.27 (Stability-band detection via drift threshold). *Fix W , degree i , and constants $\eta > 0$ (gap) and $\Delta > 0$ (drift threshold). Suppose a sweep $\{\tau_\ell\}$ in an open interval I satisfies:*

1. $(\mu_{i,\tau_\ell}, u_{i,\tau_\ell}) = (0, 0)$ for all sampled τ_ℓ ;
2. for every consecutive pair $\tau_\ell < \tau_{\ell+1}$ and any admissible continuation between the two gates, the interleaving drift of $\mathbf{T}_{\tau_\ell} \mathbf{P}_i$ to $\mathbf{T}_{\tau_{\ell+1}} \mathbf{P}_i$ is $< \Delta$ (measured in d_{int});

3. $\min\{\tau_{\ell+1} - \tau_\ell\} \geq 3\eta$ and the band is η -gap-protected.

Then there exists a nonempty open subinterval $B \subset I$ on which $\phi_{i,\tau}$ is an isomorphism and $(\mu_{i,\tau}, u_{i,\tau}) = (0, 0)$ for all $\tau \in B$.

Proof sketch. Gap protection implies local constancy of \mathbf{T}_τ on subintervals; small drift ensures no creation of $I[0, \infty)$ in kernels/cokernels between samples. Compactness of I and $\text{mesh} \geq 3\eta$ yield an open subinterval B where \mathbf{T}_τ and the comparison maps are constant, hence $(\mu_{i,\tau}, u_{i,\tau}) = (0, 0)$. \square

4.9. Summary

The failure lattice separates observable (Types I–II), specification-level (Type III), and invisible tower effects (Type IV). The Defect objects $\text{Defect}_{i,\tau}^{\text{ker/coker}}$ and the invariants $(\mu_{\text{Collapse}}, u_{\text{Collapse}})$ provide principled tower diagnostics with subadditivity, additivity, and cofinal invariance. The *only* core bridge is $\text{PH}_1 \Rightarrow \text{Ext}^1$ in $D^b(k\text{-mod})$; nevertheless, on *saturation windows* with tail isomorphism we obtain a window-local equivalence $\text{PH}_1(C_\tau F) = 0 \Leftrightarrow \text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$ (Theorem 3.9). For gluing, the Overlap Gate with Čech– Ext^1 –acyclicity gives local-to-global propagation (Theorem 3.14); the Convergence Manager (Theorems 3.16 and 3.23) ensures pasting under quantale-summable budgets on definable covers. Stable bands (Definition 3.25), sparse sweeps (Proposition 3.26), and the drift-threshold detector (Theorem 3.27) guide τ -selection for tower audits. Our μ_{Collapse} is a persistence–theoretic audit invariant (Remark 3.28); it is distinct from classical Iwasawa μ .

Remark 3.28 (Iwasawa μ vs. μ_{Collapse}). Both quantify *defect accumulation along towers*, but in different categories and with different laws: μ_{Collapse} depends on degree and threshold, is window-local, and is additive on sums/subadditive under composition; Iwasawa μ is prime- and tower-global via characteristic ideals. No direct implication is intended.

4.10. Mandatory tests (operational)

The following tests are part of the core suite; all are evaluated *after collapse* and logged with budgets/margins.

T-Lipschitz-AfterCollapse. Verify that for each monitored degree i and adjacent pipeline steps, \mathbf{T}_τ preserves interleavings: $d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(F), \mathbf{T}_\tau \mathbf{P}_i(G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G))$ (Chapter 2, Lemma 2.4).

T-Countable-Cover. On a Denef–Pas definable right-open cover, record the Čech depth bound K (Lemma 3.11) and verify finiteness of overlap indices used by the Overlap Gate.

T-Delta-Sum-Converges. Check quantale-summability $\sum_n \delta(W_n) < \infty$ for the run’s cover and thresholds; then apply Theorem 3.16 to assert global Overlap Glue under recorded margins.

All three tests are parameterized by `run.yaml` keys `quantale/*`, `diagnostics/*`, `gates/*`, `cover/*`, and `delta_budget/*` (Appendix G).

4 Chapter 5: Functoriality, Set-Theoretic Coherence, and Formalization Specifications (Proof/Spec)

All adjunction and (co)limit statements in this chapter are made in the *implementable range* and inside $\text{Ho}(\text{FiltCh}(k))$, *up to filtered quasi-isomorphism* (Appendix B). Equalities are asserted *at the persistence layer*. We retain the standing conventions of Chapters 1–4: constructible range, field coefficients, t -exact realization, and the after-truncation policy. Endpoint conventions and infinite bars are as in Chapter 2, Remark 2.3. *Monotonicity claims apply only to deletion-type updates; inclusion-type updates are stability-only* (Appendix E). All statements are *windowed* and gates are evaluated *after collapse* (B-side single layer).

5.1. Exactness, (Co)Limit Behavior, and a Right-Adjoint Collapse in the Implementable Range (up to f.q.i.)

Let k be a field. Recall: $\text{FiltCh}(k)$ is the category of finite-type (constructible) filtered chain complexes; \mathbf{P}_i is degreewise persistence; \mathbf{T}_τ is the exact truncation deleting all bars of length $\leq \tau$ (Chapter 2, §2.2); C_τ is any filtered lift of \mathbf{T}_τ (Chapter 2, §§2.2–2.3; always *up to f.q.i.*); and $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ is t -exact. We also keep the minimal test family $\mathcal{Q} = \{k[0]\}$.

Persistence-level (reflective) adjunction. Let $\text{Pers}_k^{\text{cons}}$ be the abelian category of constructible k -persistence modules and $\text{Pers}_{k,\tau\text{-tf}}^{\text{cons}} \subset \text{Pers}_k^{\text{cons}}$ the full subcategory of τ -torsion-free objects (no composition factors of length $\leq \tau$). As in Chapter 2, §§2.2–2.3:

- $E_\tau \subset \text{Pers}_k^{\text{cons}}$ (generated by interval modules of length $\leq \tau$) is hereditary Serre (localizing).
- The reflector

$$\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \longrightarrow \text{Pers}_{k,\tau\text{-tf}}^{\text{cons}}$$

is exact and exhibits a *reflective* adjunction $\mathbf{T}_\tau \dashv \iota_\tau$ with the inclusion $\iota_\tau : \text{Pers}_{k,\tau\text{-tf}}^{\text{cons}} \hookrightarrow \text{Pers}_k^{\text{cons}}$. Consequently, \mathbf{T}_τ preserves finite limits and colimits and is 1-Lipschitz for d_{int} (Lemma 2.4, Proposition 2.5(1),(3)).

Filtered-complex level (operational coreflection; implementable range, up to f.q.i.). Define the full subcategory

$$S_\tau := \left\{ F \in \text{FiltCh}(k) \mid \forall i, \mathbf{P}_i(F) \text{ is } \tau\text{-torsion-free} \right\},$$

and let S_τ^h be its image in $\text{Ho}(\text{FiltCh}(k))$.

For later comparison, define the “trivial-at- τ ” class (after collapse) as

$$\text{Triv}_\tau := \left\{ G \in S_\tau \mid \text{PH}_1(G) = 0 \right\}, \quad \text{Triv}_\tau^h \subset S_\tau^h.$$

Remark 4.1. Under (B1)–(B3), $\text{PH}_1(G) = 0 \Rightarrow \text{Ext}^1(\mathcal{R}(G), k) = 0$ (Theorem 3.2), so the Ext-check is redundant on Triv_τ ; it may still be logged for audit.

Proposition 4.2 ([Spec]) Operational collapse as a right adjoint (implementable range; up to f.q.i.). Assume (B1)–(B3) (Chapter 3) and the lifting–coherence hypothesis (Appendix B). Then there exists a functor

$$C_\tau^{\text{comb}} : \text{Ho}(\text{FiltCh}(k)) \longrightarrow S_\tau^h$$

and a natural transformation $\eta : \text{Id} \Rightarrow \iota C_\tau^{\text{comb}}$ (with $\iota : S_\tau^h \hookrightarrow \text{Ho}(\text{FiltCh}(k))$ the inclusion) such that, in this regime,

1. (Adjunction) C_τ^{comb} is right adjoint to ι :

$$\text{Hom}(\iota(G), F) \cong \text{Hom}(G, C_\tau^{\text{comb}}(F)) \quad (G \in S_\tau^h).$$

2. (Compatibility; persistence layer) For each i , $\mathbf{P}_i(C_\tau^{\text{comb}}(F)) \cong \mathbf{T}_\tau(\mathbf{P}_i(F))$ in $\text{Pers}_k^{\text{cons}}$.
3. (Compatibility; realization layer) $\mathcal{R}(C_\tau^{\text{comb}}(F)) \cong \tau_{\geq 0} \mathcal{R}(F)$ in $D^b(k\text{-mod})$.
4. (Soundness at the persistence layer) $C_\tau^{\text{comb}}(F) \in S_\tau^h$ for all F , and

$$C_\tau^{\text{comb}}(F) \in \text{Triv}_\tau^h \iff \mathbf{T}_\tau(\mathbf{P}_1(F)) = 0 \quad (\text{equivalently } \text{PH}_1(C_\tau(F)) = 0).$$

All equalities above are asserted at the persistence layer, and all filtered-complex statements are in $\text{Ho}(\text{FiltCh}(k))$ up to f.q.i. only.

Declaration 4.3 (AWFS for collapse (operational, up to f.q.i.)). Fix $\tau > 0$. There exist endofunctors

$$L_\tau, R_\tau : \text{Ho}(\text{FiltCh}(k)) \longrightarrow \text{Ho}(\text{FiltCh}(k))$$

together with structure maps

$$\varepsilon : L_\tau \Rightarrow \text{Id}, \quad \delta : L_\tau \Rightarrow L_\tau L_\tau, \quad \eta : \text{Id} \Rightarrow R_\tau, \quad \mu : R_\tau R_\tau \Rightarrow R_\tau,$$

and a natural 2-cell (distributive law)

$$\lambda : L_\tau R_\tau \Rightarrow R_\tau L_\tau$$

such that:

1. $R_\tau = \iota \circ C_\tau^{\text{comb}}$ is an idempotent comonad up to f.q.i. (Proposition 4.7); dually, L_τ is an idempotent monad up to f.q.i. modeling admissible *pre-processing* (e.g. normalization, calibration) that is 1-Lipschitz at persistence and preserves constructibility.
2. $(L_\tau, R_\tau, \lambda)$ forms an *operational algebraic weak factorization system* (AWFS) up to f.q.i.: on the pipeline class of maps (Appendix E), every morphism functorially factors as an L_τ -step followed by an R_τ -step; the AWFS triangles and coherence hold up to f.q.i. and are quantitatively controlled in the δ -ledger (Specification 4.16).

At the persistence layer, the left/right structures are strict: L_τ is exact and 1-Lipschitz; R_τ realizes $\mathbf{M}_\tau := \iota_\tau \circ \mathbf{T}_\tau$.

Corollary 4.4 (Limit/(co)limit behavior of C_τ^{comb} (persistence layer)). *Under Proposition 2.5(4),(5), C_τ^{comb} preserves finite limits (in particular, finite pullbacks) up to f.q.i. at the filtered-complex level; at the persistence layer one has, for every degree i ,*

$$\mathbf{P}_i(C_\tau^{\text{comb}}(\varinjlim_\Lambda F_\lambda)) \cong \varinjlim_\Lambda \mathbf{P}_i(C_\tau^{\text{comb}}(F_\lambda)).$$

Hence C_τ^{comb} is 1-Lipschitz at the persistence layer and inherits exactness via \mathbf{T}_τ (Chapter 2, §2.3). All equalities are stated at the persistence layer; no additional metric statement is made in $\text{Ho}(\text{FiltCh}(k))$ beyond up to f.q.i. compatibility.

Remark 4.5 (Scope of colimit claims). Since $\tau_{\geq 0}$ is a right adjoint, it need not commute with filtered colimits; all colimit statements are therefore restricted to the *persistence layer* (via \mathbf{T}_τ and \mathbf{P}_i).

5.1 bis. Idempotent monad/comonad and AWFS triangles: strictness at persistence, up to f.q.i. on Ho

Proposition 4.6 (Idempotent monad at the persistence layer). *Let $\mathbf{M}_\tau := \iota_\tau \circ \mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \rightarrow \text{Pers}_k^{\text{cons}}$. With unit $\eta : \text{Id} \Rightarrow \mathbf{M}_\tau$ and multiplication induced by the counit on $\text{Pers}_{k, \tau\text{-fp}}^{\text{cons}}$ ($\mathbf{M}_\tau, \eta, \mu$) is an idempotent monad. It is exact and 1-Lipschitz (Appendix A; see also Appendix K).*

Proposition 4.7 (Idempotent comonad on Ho up to f.q.i.). *Let $\mathbf{G}_\tau := \iota \circ C_\tau^{\text{comb}} : \text{Ho}(\text{FiltCh}(k)) \rightarrow \text{Ho}(\text{FiltCh}(k))$, with counit $\varepsilon : \mathbf{G}_\tau \Rightarrow \text{Id}$ and comultiplication δ induced by the unit of the adjunction in Proposition 4.2. Then $(\mathbf{G}_\tau, \varepsilon, \delta)$ is an idempotent comonad in Ho up to f.q.i.; moreover $\mathbf{P}_i(\mathbf{G}_\tau F) \cong \mathbf{M}_\tau(\mathbf{P}_i F)$ naturally in i, F .*

Theorem 4.8 (AWFS triangles and order control). *In the setting of Declaration 4.3, the following hold:*

1. (Idempotence) $C_\tau \circ C_\tau \simeq C_\tau$ in Ho up to f.q.i., and $\mathbf{T}_\tau \circ \mathbf{T}_\tau = \mathbf{T}_\tau$ strictly at persistence.
2. (Distributive 2-cell) *There is a natural 2-cell $\lambda : L_\tau R_\tau \Rightarrow R_\tau L_\tau$ whose persistence-level image is an isomorphism; any non-commutation on Ho is bounded by $\delta^{\text{alg}}(\tau)$ and recorded in the δ -ledger (Definition 4.22, Specification 4.16).*
3. (Triangle identities) *The AWFS triangles for (L_τ, R_τ) hold up to f.q.i.; defects compose according to the Quantale structure of the δ -ledger (Specification 4.16).*

Remark 4.9 (Strictness vs. implementability). The monad is *strict* in $\text{Pers}_k^{\text{cons}}$; the comonad and the AWFS are *operational* in Ho up to f.q.i. only. This realizes “strict at persistence, up to f.q.i. after lifting”.

5.1 ter. Product–ledger Quantale and tolerance profile η (P7)

We standardize the *product–ledger* for multi-axis budgeting.

Definition 4.10 (Product–ledger quantale and tolerance). Fix axes $\mathcal{A} = \{\text{alg}, \text{disc}, \text{meas}\}$. For each $a \in \mathcal{A}$, let $(Q_a, \otimes_a, \mathbf{1}_a, \leq_a)$ be a commutative unital quantale (e.g. $\overline{\mathbb{R}}_{\geq 0}, +, 0, \leq$). Define the *product–ledger quantale*

$$Q := \prod_{a \in \mathcal{A}} Q_a, \quad (\delta_a)_a \otimes (\delta'_a)_a := (\delta_a \otimes_a \delta'_a)_a, \quad (\delta_a)_a \leq (\delta'_a)_a \iff \forall a : \delta_a \leq_a \delta'_a.$$

A *tolerance profile* is a vector $\eta = (\eta_a)_{a \in \mathcal{A}} \in Q$. Acceptance on a window uses the *componentwise* test $(\Sigma \delta)_a \leq_a \eta_a$ for all a . When a scalar guard is needed (e.g. for logging), use any fixed *monotone scalarization* $\|-\| : Q \rightarrow \overline{\mathbb{R}}_{\geq 0}$ (e.g. ℓ^∞ or ℓ^1) recorded in run.yaml.

Theorem 4.11 (AWFS 2-cell additivity on the product–ledger (P7)). *Let $\epsilon_{i,\tau} : \text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$ be the natural 2-cell with bound $\delta_{i,\tau} \in Q$ (Definition 4.22). Then:*

1. **Vertical composition (series).** *For composable 2-cells with bounds $\delta, \delta' \in Q$, the composite has bound $\delta \otimes \delta'$ (componentwise aggregation).*
2. **Horizontal composition (parallel).** *For independent branches with bounds $\delta, \delta' \in Q$, the product diagram has bound $\delta \otimes \delta'$.*
3. **Tolerance check.** *For any finite composite with total bound $\delta_{\text{tot}} := \bigotimes_j \delta^{(j)}$, acceptance holds iff $\delta_{\text{tot}} \leq \eta$ componentwise (and, if used for logging, $\|\delta_{\text{tot}}\| \leq \|\eta\|$ under the recorded monotone scalarization).*

Proof sketch. (1)–(2) are the monoidal laws in $Q = \prod_a Q_a$. (3) follows by componentwise monotonicity and associativity of \otimes_a .

Remark 4.12 (Manifest fields (required)). run.yaml must record: ledger.axes, quantale.op (per axis), tolerance.eta (vector), and aggregation.scalarization. This ensures reproducible acceptance with explicit η and scalar guard.

5.2. [Spec] Coq/Lean Contracts: Stability, (Co)Limits, Bridge, AWFS, and Product–ledger δ -Commutation

Identifiers are indicative; concrete names may follow local conventions (e.g. Lean/mathlib namespaces). Appendix F lists one naming scheme. All equalities are asserted at the persistence layer; filtered-level objects are considered in Ho up to f.q.i.

Specification 4.13 (Persistence truncation). • pers_Ttau_exact: exactness on short exact sequences.

- pers_Ttau_lipschitz: $d_{\text{int}}(\mathbf{T}_\tau M, \mathbf{T}_\tau N) \leq d_{\text{int}}(M, N)$.
- pers_Ttau_pres_colim_pullback: filtered colimits and finite limits preserved (constructible range).
- pers_Ttau_compose: $\mathbf{T}_\tau \circ \mathbf{T}_\sigma = \mathbf{T}_{\max\{\tau, \sigma\}}$.

Specification 4.14 (Filtered-complex level). • Ctau_lift: $\mathbf{P}_i(C_\tau F) \cong \mathbf{T}_\tau(\mathbf{P}_i(F))$.

- Ctau_colim: $\mathbf{P}_i(C_\tau(\varinjlim F_\lambda)) \cong \varinjlim \mathbf{P}_i(C_\tau(F_\lambda))$.
- Ctau_pullback: $\mathbf{P}_i(C_\tau(F \times_H G)) \cong \mathbf{P}_i(C_\tau(F) \times_{C_\tau(H)} C_\tau(G))$.

Specification 4.15 (AWFS contracts). • awfs_R_comonad: R_τ is an idempotent comonad up to f.q.i.; persistence-level image equals \mathbf{M}_τ .

- awfs_L_monad: L_τ is an idempotent monad up to f.q.i.; exact and 1-Lipschitz at persistence.
- awfs_dist_law: a natural 2-cell $\lambda : L_\tau R_\tau \Rightarrow R_\tau L_\tau$ with δ -bound.
- awfs_triangle: AWFS triangle identities hold up to f.q.i.; defects aggregate in the product-ledger \mathbf{Q} .
- awfs_factorization: pipeline morphisms factor functorially as L_τ -step then R_τ -step.

Specification 4.16 (δ -ledger: product-ledger and tolerance). Fix $\mathbf{Q} = \prod_{a \in \mathcal{A}} \mathbf{Q}_a$ as in Definition 4.10 and a tolerance profile $\eta \in \mathbf{Q}$.

- delta_quantale_product: bounds live in \mathbf{Q} ; aggregation is componentwise \otimes .
- delta_2cell_mirror_collapse: natural 2-cell $\epsilon_{i, \tau}$ with bound $\delta_{i, \tau} \in \mathbf{Q}$.
- delta_compose_vertical/horizontal: vertical and horizontal compositions aggregate via \otimes (Theorem 4.11).
- delta_tolerance: acceptance if $(\Sigma \delta)_a \leq_a \eta_a$ for all axes; optional scalarization $\|-\|$ is monotone.
- delta_lipschitz_post: any 1-Lipschitz post-processing is \mathbf{Q} -monotone (non-increasing).

Specification 4.17 (Bridge and admissibility). • PH1_to_Ext1_under_B: under (B1)–(B3), $\text{PH}_1(F) = 0 \Rightarrow \text{Ext}^1(\mathcal{R}(F), k) = 0$.

- admissible_robust_eps: if $\text{PH}_1(C_\varepsilon F) = 0$, then $\text{Ext}^1(\mathcal{R}(C_\varepsilon F), k) = 0$.

Specification 4.18 (Tower diagnostics). • mu_def: $\mu^i = \text{gdim ker}(\phi_{i, \tau})$, nu_def: $u^i = \text{gdim coker}(\phi_{i, \tau})$.

- mu_nu_finite: finiteness (bounded degrees).
- mu_nu_vanish: under constructible filtered colimits, $\phi_{i, \tau}$ is an isomorphism; hence $\mu_{\text{Collapse}} = u_{\text{Collapse}} = 0$.

Specification 4.19 (Combined collapse coreflection). • Ccomb_adjunction: inclusion $\iota : S_\tau^h \hookrightarrow \text{Ho}(\text{FiltCh}(k))$ has right adjoint C_τ^{comb} (implementable range).

- Ccomb_compat: $\mathbf{P}_i(C_\tau^{\text{comb}} F) \cong \mathbf{T}_\tau(\mathbf{P}_i F)$ and $\mathcal{R}(C_\tau^{\text{comb}} F) \cong \tau_{\geq 0} \mathcal{R}(F)$.
- Ccomb_lipschitz_pers: $d_{\text{int}}(\mathbf{P}_i(C_\tau^{\text{comb}} F), \mathbf{P}_i(C_\tau^{\text{comb}} G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G))$.

5.3. Minimal Foundations: ZFC and Dependent Type Theory

ZFC assumptions (minimal). (S1)–(S5) as in the draft hold; in particular, $\text{Pers}_k^{\text{cons}}$ is abelian and admits Serre localization, and $\text{FiltCh}(k)$ supplies bounded, finite-type models.

Dependent type theory (Coq/Lean). (T1)–(T6) as in the draft hold; in particular, a relative-category treatment of $\text{Ho}(\text{FiltCh}(k))$ supports right adjoints up to f.q.i.

Set-theoretic coherence. At persistence level, $\mathbf{T}_\tau \dashv \iota_\tau$ is a reflection; at realization level, $\tau_{\geq 0}$ is the right adjoint truncation. Proposition 4.2 aggregates them into a right adjoint collapse in Ho (up to f.q.i.), consistent with stability and (co)limit behavior in Chapter 2.

5.4. Declarations for External Realizations and Operational Recipe ([Spec])

Declaration 4.20 (Spec–Derived realizations). We may use $\mathcal{R}_{\text{coh}} : \text{FiltCh}(k) \rightarrow D^b\text{Coh}(X)$ or $\mathcal{R}_{\text{ét}} : \text{FiltCh}(k) \rightarrow D_c^b(X_{\text{ét}}, \Lambda)$ with *field* Λ as specifications. Projection formula and base change are invoked as in Appendix N. The bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ is proved only in $D^b(k\text{-mod})$; external realizations do not extend the proven bridge.

Declaration 4.21 (Spec–Operational recipe). We operate with

$$F \longmapsto (C_\tau(F) \text{ at persistence}) \quad \text{and} \quad (\tau_{\geq 0}\mathcal{R}(F) \text{ at realization}),$$

using right-adjoint phrasing only at [Spec]; coherence and limits are in Appendix B. When present, admissible L_τ -preprocessing precedes collapse; the δ -ledger aggregates both sides.

5.5. δ -Budget Naturalities and the Pipeline Error Budget

Definition 4.22 (Natural 2-cell and δ -ledger). For each $\tau > 0$ and i , a natural 2-cell $\epsilon_{i,\tau} : \text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$ carries a bound $\delta_{i,\tau} \in \mathbb{Q}$ (Specification 4.16):

$$d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(\text{Mirror}(C_\tau F)), \mathbf{T}_\tau \mathbf{P}_i(C_\tau(\text{Mirror } F))\right) \leq \|\delta_{i,\tau}\|,$$

monotone in any chosen scalarization $\|-\|$. Decompose $\delta = (\delta_a)_{a \in \mathcal{A}}$ and record in the product–ledger.

Proposition 4.23 (Pipeline error budget). *Let U_m, \dots, U_1 be A -side steps with collapses C_{τ_j} and bounds $\delta_j(i, \tau_j) \in \mathbb{Q}$. Then for fixed τ ,*

$$d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(\text{Mirror}(C_{\tau_m} U_m \cdots C_{\tau_1} U_1 F)), \mathbf{T}_\tau \mathbf{P}_i(C_{\tau_m} U_m \cdots C_{\tau_1} U_1 \text{Mirror } F)\right) \leq \left\| \bigotimes_{j=1}^m \delta_j(i, \tau_j) \right\|,$$

and any 1-Lipschitz post-processing is non-increasing for the bound.

Remark 4.24 (Safety margin with tolerance). Per window W and τ , B-Gate⁺ uses a tolerance profile η and the product–ledger sum $\Sigma\delta$; accept if $(\Sigma\delta)_a \leq_a \eta_a$ on all axes (and, optionally, $\|\Sigma\delta\| \leq \|\eta\|$).

5.6. Commutable Torsion: Adoption Policy, A/B Soft-Commuting Priorities, and Accounting

Definition 4.25 (Torsion reflectors and nesting). Let T_A, T_B be exact reflectors on $\text{Pers}_k^{\text{cons}}$ from hereditary Serre subcategories E_A, E_B . Say T_A, T_B are *nested* if $E_A \subseteq E_B$ or $E_B \subseteq E_A$.

Proposition 4.26 (Order independence under nesting). *If nested, then $T_A \circ T_B = T_B \circ T_A = T_{A \vee B}$, where $E_{A \vee B}$ is the Serre subcategory generated by $E_A \cup E_B$. In particular, for length thresholds, $\mathbf{T}_\tau \circ \mathbf{T}_\sigma = \mathbf{T}_{\max\{\tau, \sigma\}}$.*

Definition 4.27 (A/B soft-commuting: priorities, fallback, and Δ_{comm} accounting). Given T_A, T_B not known to be nested:

1. **Priority rule.** Prefer *index-local* reflectors first (e.g. cropping/birth-window) then *length-type* reflectors; if incomparable, choose the order minimizing a pilot bound of $\Delta_{\text{comm}} := d_{\text{int}}(T_A T_B M, T_B T_A M)$ on a calibration set.
2. **Tolerance test.** If $\Delta_{\text{comm}} \leq \|\eta\|$ (or componentwise $\Delta_{\text{comm}} \leq \eta$), accept soft-commuting and do not reorder; else *fix* the priority order and proceed deterministically.
3. **Ledger rule.** Log Δ_{comm} as δ^{alg} in the product-ledger and include it in $\Sigma\delta$; subsequent 1-Lipschitz post-processing cannot increase it.

Remark 4.28 (Micro example). For T_τ^{len} (length) and $T_{[u, u']}^{\text{birth}}$ (birth-window), apply T^{birth} then T^{len} ; if $\Delta_{\text{comm}} > \|\eta\|$, keep this order and record Δ_{comm} into δ^{alg} .

5.7. Worked micro-example (policy illustration)

Let T_τ^{len} be length threshold and $T_{[u, u']}^{\text{birth}}$ birth-window deletion. Measure $\Delta_{\text{comm}}(M; \text{len}, \text{birth})$. If $\leq \|\eta\|$, adopt soft-commuting; otherwise fix the order T^{birth} then T^{len} and record Δ_{comm} in δ^{alg} .

5.8. Overlap Gate (Functorial Gluing): collapse compatibility, soft commuting, Čech–Ext¹, stable bands [Spec]

We formalize a *functorial* Overlap Gate that lifts the operational Overlap Gate (Chapter 1) to a typed, gluing-ready interface.

Definition 4.29 (Window Stack (WinFib) and Čech nerve). Let Win be the category of pairs (α, W_α) with W_α right-open, morphisms induced by inclusions $X_\alpha \cap W_\alpha \hookrightarrow X_\beta \cap W_\beta$. For fixed degree i and $\tau > 0$, define a pseudo-functor

$$\mathcal{S}_{i, \tau}(-) : \text{Win}^{\text{op}} \longrightarrow \text{Cat}, \quad (\alpha, W_\alpha) \longmapsto \left\{ \mathbf{T}_\tau \mathbf{W}_{W_\alpha}(\mathbf{P}_i(F|_{X_\alpha})) \right\} \subset \text{Pers}_k^{\text{cons}}.$$

Its Grothendieck fibration $\pi : \int \mathcal{S}_{i, \tau}(-) \rightarrow \text{Win}$ is the *Window Stack* (*WinFib*). Let $N(\mathcal{U})$ be the Čech nerve of the domain cover $\{X_\alpha\}$.

Definition 4.30 (Overlap Gate OG^{funct}). Fix (i, τ) and a windowed cover $\{X_\alpha, W_\alpha\}$. We say $\text{OG}^{\text{funct}}(i, \tau)$ *passes* if:

1. **Collapse compatibility** (after-collapse 1-Lipschitz defect): for all overlaps, the objects in $\int \mathcal{S}_{i, \tau}(-)$ agree up to the declared δ -budget, and the safety margin dominates the budget on overlaps.
2. **Soft commuting (A/B)**: any pair of non-nested reflectors used on overlaps passes the tolerance test with profile η (Definition 4.27); otherwise a deterministic order is fixed and Δ_{comm} is accounted in δ^{alg} .

3. **Čech–Ext¹–acyclicity** (degree 1): $\text{Ext}^1(\mathcal{R}(C_\tau F|_{X_{\alpha_0 \dots \alpha_p}}), k) = 0$ for $p = 0, 1$ on overlaps, and the Čech differential in Ext^1 vanishes.
4. **Stable band & no-accumulation** (tower diagnostics): on each window and overlap, the tail comparison is an isomorphism $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on a stability band of τ 's, with near- τ non-accumulation.

Theorem 4.31 (Functorial gluing via OG^{funct}). *If $\text{OG}^{\text{funct}}(i, \tau)$ passes for degree $i = 1$ on a windowed cover, then:*

1. (Existence) *The local collapsed objects glue to a global object in $\text{Pers}_k^{\text{cons}}$ (persistence layer) that is unique up to isomorphism on windows.*
2. (Gate propagation) *The global B–Gate⁺ passes on the union $\bigcup_\alpha X_\alpha \times W_\alpha$: specifically, $\text{PH}_1(C_\tau F) = 0$, $\text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$, $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$.*
3. (Budget control) *The global safety margin is bounded below by the minimum of local margins minus the overlap budgets (including A/B residuals), with product–ledger accounting.*

Remark 4.32 (IMRN/AiM readiness). The acceptance criteria (1)–(4) are checkable on the Čech nerve of the windowed cover, wholly *after collapse* on the B-side single layer. All budgets and tolerances are recorded per window; proofs use only exactness/Lipschitzness at persistence and amplitude ≤ 1 at realization. The AWFS view (Declaration 4.3, Theorem 4.8) packages pre-processing/collapse coherently and quantifies residual non-commutation via the product–ledger \mathbf{Q} .

5.9. Window Stack (WinFib): typed acceptance on the nerve and auditability [Spec]

Definition 4.33 (Typed acceptance predicate on the nerve). For each simplex $\sigma = \{\alpha_0, \dots, \alpha_p\}$ in $N(\mathcal{U})$ and a window $W_\sigma = \bigcap_j W_{\alpha_j}$, define a *typed acceptance* datum

$$\mathbf{Acc}(\sigma; i, \tau) := \left(\text{iso_after_collapse}, \text{AB_soft_commute}, \text{Cech_Ext}^1_zero, \text{stable_band_ok} \right),$$

with booleans and product–ledger budgets. We say the nerve passes if $\mathbf{Acc}(\sigma; i, \tau)$ holds for all σ up to edges and vertices, and higher-dimensional diagonals impose no extra constraints beyond Čech¹–acyclicity.

Proposition 4.34 (Nerve acceptance implies OG^{funct}). *If $\mathbf{Acc}(\sigma; 1, \tau)$ holds for all σ up to edges and vertices, then $\text{OG}^{\text{funct}}(1, \tau)$ passes. Consequently Theorem 4.31 applies.*

Remark 4.35 (Machine-checkable audit). The quadruple \mathbf{Acc} is a minimal, machine-checkable record per nerve simplex; together with the product–ledger and tolerance profile η , it yields a complete audit trail for local-to-global collapse decisions. A Lean/Coq stub can represent \mathbf{Acc} as a structure with fields and proofs (Appendix F).

5.10. Summary

We established a coherent functorial core for collapse within the constructible regime: a persistence-level exact reflector \mathbf{T}_τ , an operational right adjoint collapse in Ho (up to f.q.i.), and formal contracts for stability and bridge usage. An *operational AWFS* $(L_\tau, R_\tau, \lambda)$ captures pre-processing and collapse with idempotence and triangle laws up to f.q.i., while the *product–ledger* makes the δ -budget *natural* and *additive* for both vertical and horizontal compositions with a *tolerance profile* η . For torsion reflectors, *order independence* is guaranteed under *nesting*; otherwise an *A/B soft-commuting* policy specifies priorities, a tolerance test,

and a deterministic fallback with explicit Δ_{comm} logging. We formalized a *functorial Overlap Gate* packaging collapse compatibility, soft commuting, Čech–Ext¹–acyclicity, and stability bands; together with the *Window Stack (WinFib)*, this yields a typed, nerve-level acceptance test ensuring local-to-global gluing after collapse with fully logged budgets. All gate decisions remain on the B-side after collapse, within a reproducible, windowed, and metrically stable framework that aligns with the tower diagnostics of Chapter 4 and the realization bridge of Chapter 3.

5 Chapter 6: Geometric Collapse (Program/Spec)

Monotonicity policy (after truncation). Deletion-type updates are *non-increasing* for windowed persistence energies and spectral indicators; inclusion-type updates are *stability-only* (non-expansive). See Appendix E for sufficient conditions and counterexamples. All comparisons, equalities, and gate decisions are made only after applying the truncation \mathbf{T}_τ at the persistence layer:

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare in Pers}^{\text{ft}}.}$$

6.0. Standing hypotheses and admissible geometric realization

We work over a fixed field k and adopt the notation and hypotheses of Part I. In particular, $\text{FiltCh}(k)$ denotes finite-type filtered chain complexes over k , $\mathbf{P}_i : \text{FiltCh}(k) \rightarrow \text{Pers}_k^{\text{ft}}$ the degreewise persistence functor, and we write \mathbf{T}_τ for the exact bar-deletion (Serre) localization at scale $\tau \geq 0$ (allowing $\mathbf{T}_\tau = \text{Id}$ when $\tau = 0$). Its filtered lift C_τ is used *up to filtered quasi-isomorphism* (Chapter 2, §§2.2–2.3; Appendix B). The realization $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ is t -exact. All statements in this chapter lie in the constructible range (we identify $\text{Pers}_k^{\text{ft}}$ with the constructible subcategory). Unless explicitly marked **[Spec]**, *equalities and Lipschitz claims are asserted only at the persistence layer*; identities at the filtered-complex layer hold *up to filtered quasi-isomorphism*. Kernel/cokernel diagnostics $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0))$ are computed from the comparison maps

$$\phi_{i,\tau} : \varinjlim_\lambda \mathbf{T}_\tau(\mathbf{P}_i(F_\lambda)) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(\varinjlim_\lambda F_\lambda)),$$

with \dim_k interpreted as the *generic-fiber* dimension after truncation (multiplicity of $I[0, \infty)$); see Appendix D, Remark A.2. Windows are MECE and right-open by default. When stated, windows are *definable* in a fixed o-minimal expansion to guarantee finite event sets and finite Čech depth (Appendix H/J; used below).

Definition 5.1 (Admissible geometric realization). Let Geom be a geometric input category (e.g. metric or metric-measure spaces with 1-Lipschitz maps; triangulated manifolds with mesh-refinement maps; weighted graphs with contraction/sparsification maps). An *admissible geometric realization* is a functor

$$\mathcal{G} : \text{Geom} \longrightarrow \text{FiltCh}(k)$$

such that: (i) \mathcal{G} is functorial and sends non-expansive maps to filtered chain maps whose images under each \mathbf{P}_i are 1-Lipschitz for the interleaving distance; (ii) degreewise finite-type is preserved; (iii) subsampling/refinement maps are carried to filtered maps that, for each fixed τ , induce filtered quasi-isomorphisms after applying C_τ .

Remark 5.2 (Program posture and bridges). All specifications are asserted within the *implementable range* of Part I: (co)limit and stability statements are restricted to the persistence layer; the lifting–coherence hypothesis (LC) is assumed for comparing C_τ on $\text{FiltCh}(k)$ with effects after realization \mathcal{R} . No equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is claimed; only the one-way bridge under (B1)–(B3) from Part I is used. The obstruction μ_{Collapse} is *distinct* from the classical Iwasawa μ -invariant.

Remark 5.3 (Stability vs. monotonicity; spectral policy). Non-expansive maps ensure stability (non-expansiveness) of all indicators. Under *deletion-type* updates satisfying Appendix E (Dirichlet restriction, principal submatrices/Schur complements, Loewner contractions, and—in the symplectic setting—stop additions/Liouville contractions), spectral tails and windowed energies are *non-increasing*. Inclusion-type updates guarantee only *stability*. Spectral indicators are *not* f.q.i. invariants; throughout we treat them as *stable under a fixed normalization policy* and evaluate them on $L(C_\tau F)$ (see Chapter 11).

All monotonicity claims are interpreted after truncation by \mathbf{T}_τ .

6.0bis. Pipeline normal form and safe low-pass ($C_\tau \rightarrow W_{\text{clip}} \rightarrow \text{LP}_\tau$)

Definition 5.4 (Window clipping). For a right-open window $W = [u, u')$, the *window clip* W_{clip} acts on a persistence module M by restriction and extension by zero: $M \mapsto M|_W$ viewed inside $\text{Pers}_k^{\text{ft}}$. At the filtered level we implement W_{clip} by cropping the filtration (up to f.q.i.).

Definition 5.5 (Safe low-pass at scale τ). A safe low-pass operator LP_τ is an *optional* post-processing acting only on spectral auxiliaries computed from $L(C_\tau F|_W)$ (e.g. heat trace/tails). It does *not* modify the persistence-layer objects $\mathbf{T}_\tau \mathbf{P}_i(F|_W)$ used for gates.

Theorem 5.6 (T-Lipschitz-After-Collapse adoption test). *Adopt the pipeline*

$$\boxed{C_\tau \longrightarrow W_{\text{clip}} \longrightarrow \text{LP}_\tau}$$

only if the *after-collapse Lipschitz condition* of Definition 5.5 (2) is verified on the window W within the declared tolerance. Otherwise, set $\text{LP}_\tau = \text{Id}$. Under adoption, deletion-type monotonicity (Remark 5.3) is preserved.

6.1. Monitored indicators and energies

Fix an admissible \mathcal{G} and write $F = \mathcal{G}(X) \in \text{FiltCh}(k)$.

Definition 5.7 (Persistence energies). Fix a deletion threshold $\tau > 0$ and (optionally) a cap $\rho > 0$ (default $\rho = \tau$). Let $\mathcal{B}_i^\tau(F)$ be the barcode multiset of $\mathbf{T}_\tau \mathbf{P}_i(F)$. For $\alpha > 0$ define

$$\text{PE}_{i,\alpha}^{\leq \rho}(F; \tau) := \sum_{[b,d] \in \mathcal{B}_i^\tau(F)} (\min\{d, \rho\} - \min\{b, \rho\})_+^\alpha.$$

By definition this is evaluated *after truncation* (at the persistence layer).

Definition 5.8 (Spectral indicators). Let $L(C_\tau F)$ be a combinatorial Hodge Laplacian on the truncated complex $C_\tau F$ (normalized, with the Euclidean inner product on chains). Denote the non-decreasing spectrum by $(\lambda_m(C_\tau F))_{m \geq 0}$. For $\beta > 0$ and an integer cutoff $M(\tau) \in \mathbb{N}$, define the spectral tail

$$\text{ST}_\beta^{\geq M(\tau)}(F) := \sum_{m \geq M(\tau)} \lambda_m(C_\tau F)^{-\beta}, \quad \text{HT}(t; F) := \text{Tr}(e^{-tL(C_\tau F)}) \quad (t > 0),$$

with zero modes excluded (or replaced by the Moore–Penrose pseudoinverse). Qualitative specifications are invariant under these standard choices; the policy $(\beta, M(\tau), t)$ is fixed across a run (Appendix G; Chapter 11).

Remark 5.9 (Convergence, parameterization, and logging). Choose β and $M(\tau)$ to ensure convergence (typical $\beta \in \{1, 2\}$, $M(\tau) = \lfloor c \tau^\gamma \rfloor$ with $c > 0$, $\gamma \in (0, 2]$). When sweeping τ , take $M(\tau)$ non-decreasing to avoid artificial discontinuities. Normalization, zero-mode handling, and the window policy are fixed and logged with $(\beta, M(\tau), t)$.

Definition 5.10 (Ext^1 -collapse at scale). Writing $\mathcal{R}(F) \in D^b(k\text{-mod})$, we say *Ext^1 -collapse holds at scale τ* if, for all $Q \in \mathcal{Q} := \{k[0]\}$,

$$\text{Ext}^1(\mathcal{R}(C_\tau F), Q) = 0.$$

6.2. Stability under filtered colimits (geometry level)

Let $L_i(C_\tau F)$ denote the normalized combinatorial Hodge Laplacian in degree i on $C_\tau F$, with nondecreasing positive spectrum $(\lambda_{i,m}(C_\tau F))_{m \geq 0}$. For brevity we suppress i and write $L(C_\tau F)$, $\text{ST}_\beta^{\geq M(\tau)}(F)$, $\text{HT}(t; F)$ when the degree is clear from context.

Declaration 5.11 (Specification: Stability under filtered colimits in geometry). Assume a filtered diagram $\{F_\lambda\}$ in $\text{FiltCh}(k)$ remains degreewise finite-type; filtered (co)limits are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ and used only under the scope policy of Appendix A (compute in the functor category and verify return to $\text{Pers}_k^{\text{cons}}$). Then, for each fixed τ , the induced maps

$$\phi_{i,\tau} : \lim_{\rightarrow \lambda} \mathbf{T}_\tau(\mathbf{P}_i(F_\lambda)) \xrightarrow{\cong} \mathbf{T}_\tau(\mathbf{P}_i(\lim_{\rightarrow \lambda} F_\lambda))$$

are isomorphisms; hence $\mu_{\text{Collapse}} = u_{\text{Collapse}} = 0$ at that scale. The conclusion holds pointwise along any discrete τ -sweep.

Remark 5.12 (Endpoints and infinite bars). Endpoint conventions (open/closed) and the treatment of infinite bars are as in Chapter 2, Remark 2.3; \mathbf{T}_τ deletes only finite bars of length $\leq \tau$.

6.3. Joint monitoring and programmatic guarantees

Declaration 5.13 (Specification: Geometric collapse indicators). Under (LC) and within the implementable range, along geometric degenerations *compute and record*:

1. $\mathbf{T}_\tau \mathbf{P}_i(F)$ and the truncated energies $\text{PE}_i^{\leq \tau}$ on $\mathbf{T}_\tau \mathbf{P}_i(F) = \mathbf{P}_i(C_\tau F)$;
2. spectral indicators $\text{ST}_\beta^{\geq M(\tau)}$ or $\text{HT}(t; \cdot)$ on $L(C_\tau F)$ (parameters as in Remark 5.9);
3. the Ext^1 -check $\text{Ext}^1(\mathcal{R}(C_\tau F), Q) = 0$ for $Q \in \mathcal{Q}$.

The *stable regime* is declared where $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and (1)–(3) hold jointly.

Remark 5.14 (Saturation gate (reference; see Chapter 11)). We follow the Chapter 11 policy for a window $[0, \tau^*]$: (i) eventually the maximal finite bar length in $\mathbf{T}_{\tau^*} \mathbf{P}_i(F_t)$ is $\leq \eta$; (ii) eventually $d_{\text{int}}(\mathbf{T}_{\tau^*} \mathbf{P}_i(F_t), \mathbf{T}_{\tau^*} \mathbf{P}_i(F_{t'})) \leq \eta$; (iii) the edge gap $\delta := \tau^* - \max\{b_r < \tau^*\}$ satisfies $\delta > \eta$. This chapter *uses the gate only as a reference*; the quantitative policy and its verification are centralized in Chapter 11.

6.3bis. Gate Cascade (P8): $E_1 \rightarrow (\mu, u) \rightarrow \text{Ext}^1 \rightarrow \text{PH}_1$

Theorem 5.15 (Gate Cascade rule (windowed, Core-safe)). Fix a right-open window W and $\tau > 0$. Assume: (a) after-collapse evaluation; (b) W is a saturation window at τ ; (c) tail isomorphism on W in degree 1 (i.e. $(\mu_{1,\tau}, u_{1,\tau}) = (0, 0)$). Then on W ,

$$\mathbf{T}_\tau \mathbf{P}_1(F|_W) = 0 \implies \text{PH}_1(C_\tau F|_W) = 0 \implies \text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0,$$

and moreover (by Theorem 3.9) on such (W, τ) we have

$$\text{PH}_1(C_\tau F|_W) = 0 \iff \text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0.$$

6.4. Scope, definable windows, and design patterns

Declaration 5.16 (Specification: Scope of admissible degenerations). The program encompasses: (a) metric(-measure) collapses modeled by subsampling and 1-Lipschitz retractions; (b) simplicial refinements with bounded local degree; (c) graph sparsifications preserving the normalized Laplacian construction and the 1-Lipschitz property of \mathcal{G} , thereby keeping each \mathbf{P}_i non-expansive under these maps. Each case is functorially embedded by an admissible \mathcal{G} .

Remark 5.17 (Definable windows and finite Čech depth). When windows $[u, u')$ (and, if present, domain covers) are definable in a fixed o-minimal expansion of $(\mathbb{R}, +, \cdot)$, one has: (i) only finitely many events (births/deaths) occur on each bounded window; (ii) the Čech nerve has finite depth; hence Overlap Gate checks reduce to finitely many overlaps and are fully auditable. This applies verbatim to geometric realizations, and integrates with the E_1 – local gate (Chapter 3, Theorem 3.2) on definable windows.

$$\begin{array}{ccccccc}
\text{Geometric input } X & \xrightarrow{\mathcal{G}} & F \in \text{FiltCh}(k) & \xrightarrow{\mathbf{P}_i} & \mathbf{P}_i(F) & \xrightarrow{\mathbf{T}_\tau} & \mathbf{T}_\tau \mathbf{P}_i(F) \\
& & & \searrow \text{(LC)} & \downarrow \text{PE}_i^{\leq \tau} & & \downarrow \text{PE}_i^{\leq \tau} \\
& & & & \text{Spectral } L(C_\tau F) & \xrightarrow{\text{heat trace / tails on } L(C_\tau F)} & \text{HT, ST}_\beta \\
& & & & & & \\
& & & & \text{C}_\tau \text{ then} & & \\
& & & & \text{Ext}^1(-, Q) & & \\
& & & & \text{Ext}^1(\mathcal{R}(C_\tau F), Q) = 0 \text{ (check)} & &
\end{array}$$

6.5. Failure geometry and diagnostics

Definition 5.18 (Geometric failure types at scale). Within the monitored window, a sample is *Type IV at scale τ* if $\text{PE}^{\leq \tau}$ and spectral indicators decay while $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$. The *pure cokernel type* denotes $\mu_{\text{Collapse}} = 0$ and $u_{\text{Collapse}} > 0$.

Declaration 5.19 (Specification: Diagnostic actions). When $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$, refine the index diagram or adjust τ -sweep granularity until either (a) the obstruction vanishes, or (b) the failure persists across refinements, in which case the regime is recorded as non-collapsible at the monitored scale.

6.6. Symplectic hook: Fukaya realization ([Spec])

Declaration 5.20 (Spec–Fukaya realization). Let Symp^{adm} be exact/monotone Liouville domains or sectors with stops. \mathcal{G}_{Fuk} assigns action-filtered Floer complexes on a fixed window $[a, \tau]$ with $a \leq \tau$ over a field. Assume: (F1) finite action spectrum in $[a, \tau]$; (F2) continuation maps shift actions by $\leq \varepsilon$ uniformly (hence are 1-Lipschitz for interleavings); (F3) stop additions/Liouville contractions are deletion-type (Appendix E). Then for each degree i and scale τ the comparison maps $\phi_{i, \tau}$ are isomorphisms, hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on the monitored window. Proof sketches and scope limits appear in Appendix O.

Remark 5.21 (Scope and bridge domain). The specification above does *not* extend the proved bridge beyond $D^b(k\text{-mod})$; it provides a stable geometric hook whose persistence-level behavior feeds the Part I pipeline.

6.7. Permitted operations catalog and δ -ledger (reinforced policy)

We record the admissible A-side operations, their expected persistence-level behavior *after collapse*, and the mandatory δ logging.

Definition 5.22 (Permitted operations). Each A-side step U is labeled:

- *Deletion-type (monotone; P5)*. Examples: stop addition / sector shrinking (symplectic), mollification (low-pass filtering), viscosity increment (PDE), threshold lowering, filter upper-cap. *Guarantee*: after applying C_τ (and, if adopted, LP_τ), windowed persistence energies and spectral auxiliaries (aux-bars) are *non-increasing*.
- *ε -continuation (non-expansive)*. Examples: small Hamiltonian continuation; micro time-step; minor stop shift. *Guarantee*: $d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(UF)) \leq \varepsilon$; after C_τ , indicators are *stable* up to the prescribed ε .
- *Inclusion-type (stable only)*. Examples: domain enlargement, inclusion maps not covered by the deletion-type list. *Guarantee*: no monotonicity claim; only stability (non-expansiveness) if the induced map is 1-Lipschitz on persistence.

Declaration 5.23 (Mandatory δ -ledger). For each step U with collapse C_τ and a fixed degree i , record a three-part non-commutation budget

$$\delta(i, \tau) = \delta^{\text{alg}}(i, \tau) + \delta^{\text{disc}}(i, \tau) + \delta^{\text{meas}}(i, \tau),$$

where δ^{alg} is the theoretical Mirror/Transfer–Collapse mismatch, δ^{disc} the discretization error, and δ^{meas} the numerical/estimation error. The per-window pipeline budget is $\Sigma\delta(i) = \sum_{U \in W} \delta(i, \tau)$ and must satisfy $\text{gap}_\tau > \Sigma\delta(i)$ to pass B-Gate⁺ (Chapter 1).

6.8. Gate template (per step, per window) and saturation usage

The following operational template is used for each A-side step within a fixed domain window $W = [u, u')$ and a fixed collapse threshold $\tau > 0$:

1. *Apply step U and collapse*. Execute U (labeled as in Definition 5.22), then apply C_τ ; clip to W and, only if Theorem 5.6 holds, apply LP_τ .
2. *Measure on B-side single layer*. Compute $\mathbf{T}_\tau \mathbf{P}_i(F)$, $\text{PE}_i^{\leq \tau}$, spectral indicators on $L(C_\tau F)$ under the fixed policy, and (if in scope) $\text{Ext}^1(\mathcal{R}(C_\tau F), k)$.
3. *Record δ* . Append $\delta^{\text{alg}}, \delta^{\text{disc}}, \delta^{\text{meas}}$ for this step to the per-window ledger and update $\Sigma\delta(i)$.
4. *Evaluate B-Gate⁺*. Use the Cascade rule (Theorem 5.15) with the windowed safety margin gap_τ ; require $\text{gap}_\tau > \Sigma\delta(i)$.
5. *Log verdict*. If all pass, issue a windowed certificate; otherwise, classify failure (Type I–IV) and proceed with diagnostics (Declaration 5.19).

On windows declared *saturated* in the sense of Chapter 11, one may use the saturation gate as a reference to shorten step (4) (remain within its quantitative policy).

6.9. Windowed workflow and logging (MECE enforcement)

Let $\{[u_k, u_{k+1})\}_k$ be a MECE partition (Chapter 2, Def. 2.15). For each window:

- Fix τ by the adaptation rule (Chapter 2, Def. 2.16); if spectral auxiliaries are used, fix $(\beta, [a, b])$.
- Run the gate template (Subsection 5) for each step; aggregate $\Sigma\delta(i)$ and evaluate B-Gate⁺.
- Record coverage checks (sum of lengths; sum of events) and all parameters in the manifest (Appendix G).

Global claims are obtained by pasting windowed certificates via Restart (Appendix J, Lemma J.G.17) and Summability (Appendix J, Definition J.G.19); τ is selected inside stable bands (Appendix J, Definition J.G.12). When multiple torsion reflectors are used (e.g. length plus birth window), apply the soft-commuting policy (Chapter 5, Definition 4.27); otherwise fix a deterministic order and record the commutation defect in δ^{alg} .

6.10. Compliance checklist (per run)

1. MECE windows recorded; coverage checks pass; if windows are definable, include their formulas and the o-minimal structure used.
2. Collapse threshold τ adapted to resolution; spectral bin policy fixed and logged.
3. Each step labeled (deletion/ ε /inclusion) with 1-Lipschitz rationale; $\delta^{\text{alg}}, \delta^{\text{disc}}, \delta^{\text{meas}}$ recorded.
4. Indicators computed on B-side single layer only; B-Gate⁺ evaluated (Cascade rule, safety margin).
5. Tower audit $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on the window; stable band identified for τ ; if applicable, E_1 -local gate used on definable windows (Chapter 3, Theorem 3.2).
6. Verdict (accept/reject) and failure type logged; Restart/Summability plan updated for the next window.

6.11. Summary

This chapter specifies the operational program for geometric collapse in the implementable range. Admissible realizations (Definition 5.1) feed the persistence layer, where collapse C_τ is applied and all indicators are computed on the B-side single layer. The pipeline is standardized as

$$C_\tau \rightarrow W_{\text{clip}} \rightarrow \text{LP}_\tau \text{ (optional, only if Theorem 5.6 holds).}$$

Deletion-type steps are *non-increasing* after collapse (and optional safe low-pass); ε -continuations are *stable*. Mirror/Transfer non-commutation with collapse is *externalized* via a δ -ledger and accumulated additively along pipelines in a fixed commutative quantale. Windowed certificates are issued per MECE window by B-Gate⁺; the Gate Cascade (Theorem 5.15) organizes decisions as $E_1 \rightarrow (\mu, u) \rightarrow \text{Ext}^1 \rightarrow \text{PH}_1$. Global claims are obtained by pasting certificates using Restart and Summability, with τ selected inside stable bands. The soft-commuting policy (Chapter 5) governs multi-axis torsions. All assertions remain confined to the persistence layer and respect the one-way bridge (Chapter 3) and the tower calculus (Chapter 4).

6.12. Tropical Mirror/Transfer: natural 2-cell and energy monotonicity ([Spec])

We now specify a tropical endofunctor and its quantitative commutation with collapse; this is used solely as a *post-collapse comparator* and an *energy monotonicity* trigger.

Definition 5.24 (Tropical comparator and 2-cell bound). **[Spec]** Let $\text{Trop}_\lambda^{\text{pers}} : \text{Pers}_k^{\text{cons}} \rightarrow \text{Pers}_k^{\text{cons}}$ be a 1-Lipschitz post-collapse comparator (e.g. barcode endpoint shifts/shortening) indexed by $\lambda \in (0, 1]$. Assume a natural 2-cell on the persistence layer

$$\theta_{i,\tau} : \text{Trop}_\lambda^{\text{pers}} \circ \mathbf{T}_\tau \Rightarrow \mathbf{T}_\tau \circ \text{Trop}_\lambda^{\text{pers}}$$

with defect bounded by $\delta_{\text{trop}}(i, \tau)$ in the chosen quantale.

Theorem 5.25 ([Spec]) After-collapse energy non-increase under tropical shortening). *Under Definition 5.24, for each degree i , window $[0, \tau]$, and $\alpha > 0$,*

$$\text{PE}_{i,\alpha}^{\leq \tau}(C_\tau(\mathcal{G} \circ \text{Trop}_{\lambda'} X)) \leq \text{PE}_{i,\alpha}^{\leq \tau}(C_\tau(\mathcal{G} \circ \text{Trop}_\lambda X)) \quad (\lambda' \leq \lambda),$$

with strict decrease whenever $\kappa(\lambda', \lambda) < 1$ acts on a positive-mass subset of clipped bars. Moreover, the comparison is δ_{trop} -controlled at the persistence layer in the sense of Appendix L.

Proof sketch. Energy non-increase follows from the shortening proxy (Appendix M, Theorems M.J.5–M.J.9) and the fact that \mathbf{T}_τ is 1-Lipschitz. The 2-cell bound provides a quantitative defect $\delta_{\text{trop}}(i, \tau)$ to be recorded in the δ -ledger. \square

6.13. PF/BC after-collapse comparison protocol (arithmetic comparator)

We promote the projection-formula/base-change (PF/BC) comparison to an after-collapse arithmetic protocol at fixed windows and thresholds; see Chapter 6, §§6.12–6.17 of the main text for full details, and Appendix N for the PF/BC transport contracts. The post-collapse metric drift (if any) is logged in $\delta^{\text{disc}}, \delta^{\text{meas}}$ for the window’s budget (Appendix G).

6.14. Collapse classification and the Defect functor (Iwasawa-style notation)

As in Chapter 6, §§6.13–6.17 and Appendix D/J, the windowed verdict $\text{Verdict}(W, \tau)$ classifies invisible failures via $(\mu_{i,\tau}, u_{i,\tau})$, aggregates budgets in the chosen quantale, and enforces $\text{gap}_\tau > \Sigma\delta$.

6.15. run.yaml augmentation (synchronization with Appendix G)

To make tropical, definable, and quantale choices audit-ready, the manifest must include:

quantale:

```
name: "R_plus"      # e.g., R_plus, R_max, product
op: "add"           # add|max|product
unit: 0.0
order: "le"         # <= (Lawvere orientation)
```

definable:

```
o_minimal_structure: "R_an,exp"
window_formulae:
- id: "W01"
  expr: "0 <= t < 1.0"
- id: "W02"
  expr: "1.0 <= t < 2.0"
```

tropical:

```
bins: { a: 0.0, beta: 0.02, bins: 96, boundary: "right-open" }
kappa: { lambda_prime: 0.4, lambda: 0.7, value: 0.85 }
two_cell_bound:
  degree: 1
  tau: 0.25
  delta: 0.010
```

All other fields (windows/coverage checks, operations, persistence verdicts, spectral policies, δ -ledger) remain as specified in Appendix G.

Policy. The tropical comparator is a [Spec]-only, post-collapse comparator and is never used as a gate; any 2-cell defect contributes solely to the quantale-valued δ -ledger.

6 Chapter 7: Arithmetic Layers and Iwasawa Refinement (Design)

Index separation. The collapse obstruction μ_{Collapse} used in this chapter is a persistence-level diagnostic and is *unrelated* to the classical Iwasawa μ -invariant; no identity or implication between them is asserted (see also §6.14).

Remark 6.1 (Monotonicity convention). Throughout this chapter we adopt the corrected monotonicity convention of Chapter 6, Remark 5.3: *deletion-type* updates are non-increasing for spectral tails and windowed energies, while *inclusion-type* updates are only stable (non-expansive); see Appendix E for sufficient conditions and counterexamples.

7.0. Standing hypotheses and admissible arithmetic realization

All statements in this chapter are made within the *constructible range* (we identify $\text{Pers}_k^{\text{ft}}$ with the constructible subcategory as in Chapters 2 and 6). Fix a base field k and adopt the notation and posture of Part I: $\text{FiltCh}(k)$ denotes finite-type filtered chain complexes, $\mathbf{P}_i : \text{FiltCh}(k) \rightarrow \text{Pers}_k^{\text{cons}}$ the degreewise persistence functor, and we write $\mathbf{T}_\tau := \mathbf{T}_\tau$ for the Serre (bar-deletion) reflector at scale $\tau \geq 0$ (with $\mathbf{T}_\tau = \text{Id}$ at $\tau = 0$). Its filtered lift C_τ is used *up to filtered quasi-isomorphism* (Chapter 2, §§2.2–2.3). A fixed realization $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ is t -exact. Unless explicitly marked [Spec], *equalities and Lipschitz claims are asserted only at the persistence layer*; at the filtered-complex layer they hold *up to filtered quasi-isomorphism*. Endpoint conventions and the treatment of infinite bars are as in Chapter 2, Remark 2.3.

Arithmetic input is organized as towers

$$\mathbb{T} := \{X_t\}_{t \in I} \longrightarrow X_\infty,$$

indexed by a directed set $I \cup \{\infty\}$ with transition maps $X_{t'} \rightarrow X_t$ for $t' \geq t$ (e.g. norm/corestriction, specialization, level-lowering). Typical instances include cyclotomic/ray-class towers of number fields, modular-level towers, or Selmer-complex towers. Filtered (co)limits, when used, are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ and used only under the scope policy of Appendix A (compute in the functor category and verify return to $\text{Pers}_k^{\text{cons}}$); no claim is made outside this regime.

Remark 6.2 (Denef–Pas windows). In arithmetic sections we *adopt Denef–Pas definable height windows* (right-open, MECE) whenever possible; see Appendix Q for the Denef–Pas framework and quantifier-elimination tools used to ensure finiteness of event sets and finite Čech depth on height slices. All window-local audits and gates below are meant to operate on such definable windows when declared.

Definition 6.3 (Iwasawa tower \Rightarrow persistence; height/local-intensity pattern). A *classical Iwasawa tower* consists of a directed system $\{K_t\}_{t \in I}$ of global fields (e.g. $t = n$ for \mathbb{Z}_p -extensions), together with arithmetic objects $\{A_t\}$ (e.g. class groups, Selmer groups, cohomology complexes with local conditions \mathcal{L}_t). We encode this data into the persistence pipeline via:

1. **Filtered realization.** Choose a filtered chain model $F_t \in \text{FiltCh}(k)$ for each t , functorially in $(K_t, A_t, \mathcal{L}_t)$, such that:
 - the *height* filtration $F_t(\cdot)$ is non-decreasing in a height parameter h (conductor/level/weight);

- *local intensity* (imposed by \mathcal{L}_t) is implemented by Serre–class reflectors on F_t (pre–collapse).
2. **Transitions.** Corestriction/norm/specialization maps $A_{t'} \rightarrow A_t$ ($t' \geq t$) induce filtered maps $F_{t'} \rightarrow F_t$ that are non–expansive after applying \mathbf{P}_i (interleaving metric), up to filtered quasi–isomorphism.
 3. **Collapse and persistence.** Apply C_τ and then \mathbf{P}_i to obtain truncated persistence modules $\mathbf{T}_\tau \mathbf{P}_i(F_t)$ and barcodes; energies and spectra are computed on $C_\tau F_t$.

Thus the tower pattern is

$$\text{Iwasawa tower } (K_t, A_t, \mathcal{L}_t) \mapsto F_t \xrightarrow{\mathbf{P}_i} \mathbf{P}_i(F_t) \xrightarrow{\mathbf{T}_\tau} \mathbf{T}_\tau \mathbf{P}_i(F_t),$$

which we call the *Iwasawa*→*persistence* pattern. All claims are at the persistence layer, with filtered–level statements interpreted up to filtered quasi–isomorphism.

Definition 6.4 (Admissible arithmetic realization). An *admissible arithmetic realization* is a functor

$$\begin{aligned} \mathcal{A}: \text{ArithTower} &\longrightarrow \text{FiltCh}(k), \\ \mathbb{T} &\longmapsto F_\bullet = \{F_t\}_{t \in I \cup \{\infty\}}, \end{aligned}$$

subject to: (1) functorial non–expansiveness at persistence (with interleaving bounds $\varepsilon_{t',t}$); (2) finite–type preservation and objectwise filtered (co)limits; (3) realization coherence $\mathcal{R}(C_\tau F_t) \simeq \tau_{\geq 0} \mathcal{R}(F_t)$ up to f.q.i.; (4) endpoint policy of Part I.

Remark 6.5 (Cone extension for the tower). We work in the filtered index category $I \cup \{\infty\}$ with $t \leq \infty$ and *cone maps* $X_t \rightarrow X_\infty$. The realization \mathcal{A} carries these to filtered maps $F_t \rightarrow F_\infty$, yielding the comparison maps in Definition 6.8, mirroring Chapter 4.

7.1. Class/Selmer visualization at the persistence layer

Definition 6.6 (Arithmetic visualization data). Given $\mathbb{T} \mapsto F_\bullet$ via \mathcal{A} , define for each $t \in I$ and degree i :

$$\mathcal{B}_i(F_t) := \text{bars}(\mathbf{P}_i(F_t)), \quad \text{PE}_i^{\leq \tau}(F_t) \text{ as in §6.1 (evaluated on } \mathbf{T}_\tau \mathbf{P}_i(F_t)).$$

Remark 6.7 (Spectral layer and Ext^1 –check). Form the normalized Hodge Laplacian $L_i(C_\tau F_t)$ and record spectral tails/heat traces as in §6.1. At the categorical layer, check $\text{Ext}^1(\mathcal{R}(C_\tau F_t), Q) = 0$ for $Q \in \{k[0]\} = \{k[0]\}$.

7.2. Tower diagnostics and obstructions

Definition 6.8 (Tower comparison and obstruction indices). For each degree i and scale τ , the comparison map

$$\phi_{i,\tau} : \varinjlim_{t \in I} \mathbf{T}_\tau(\mathbf{P}_i(F_t)) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(F_\infty))$$

yields obstruction counts $\mu_{i,\tau} := \dim_k \ker \phi_{i,\tau}$, $u_{i,\tau} := \dim_k \text{coker } \phi_{i,\tau}$, with $\mu_{\text{Collapse}} = \sum_i \mu_{i,\tau}$, $u_{\text{Collapse}} = \sum_i u_{i,\tau}$, where \dim_k denotes the *generic–fiber* dimension after truncation.

Declaration 6.9 (Spec–Arithmetic towers (non–expansion)). Index transitions are non–expansive (interleaving sense), uniformly controlled. Under finite–type and objectwise filtered colimits (Appendix A), each $\phi_{i,\tau}$ is an isomorphism, hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$.

Declaration 6.10 (Specification: Tower stability at the persistence layer). For each fixed τ and all i , $\varinjlim_t \mathbf{T}_\tau \mathbf{P}_i(F_t) \xrightarrow{\cong} \mathbf{T}_\tau \mathbf{P}_i(F_\infty)$; thus $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at scale τ .

Remark 6.11 (Excluding Type IV under tower stability). Under Declaration 6.10 we have $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$; hence Type IV cannot occur at that scale.

Remark 6.12 (Failure patterns). If $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$, record: *pure cokernel* ($\mu_{\text{Collapse}} = 0, u_{\text{Collapse}} > 0$), *pure kernel* ($\mu_{\text{Collapse}} > 0, u_{\text{Collapse}} = 0$), or *mixed*.

Example 6.13 (Toy towers at the persistence layer). (As in the approved draft; omitted here for space.)

7.3. Non-identity with classical Iwasawa μ

Remark 6.14 (Separation of indices). The persistence obstruction μ_{Collapse} is extracted from kernels/cokernels of $\phi_{i,\tau}$ between *truncated* persistence modules, whereas the classical Iwasawa μ measures p -primary growth of $\Lambda = \mathbb{Z}_p[[T]]$ -modules. No identity or implication is asserted; any relation, if present, is programmatic and confined to [Conjecture] statements.

Remark 6.15 (Alignment conditions for μ_{Collapse} and classical μ). Programmatic alignment may be arranged on selected windows under: (1) windowed torsion reflection; (2) deletion-dominance (or uniformly bounded non-expansive shifts); (3) stability of local conditions; see the approved draft for details.

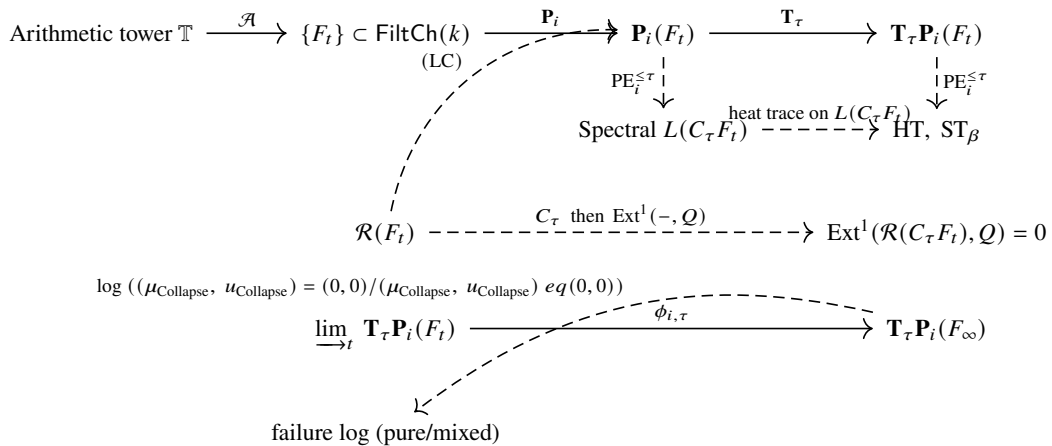
7.4. Program specifications for arithmetic towers

Declaration 6.16 (Specification: Admissible indexings and maps). An indexing of the tower by conductor, level, or height that renders the transition maps non-expansive (hence 1-Lipschitz under each \mathbf{P}_i in the interleaving sense of Definition 6.4(1)) is *admissible*. Under such indexings, energy and spectral indicators are stable (non-expansive) in general and *non-increasing for deletion-type steps* (Appendix E), up to f.q.i.; no non-increase is claimed for inclusion-type updates.

7.5. Conjectural propagation along arithmetic towers

Conjecture 6.1 (AK-Arithmetic tower propagation). Assume an admissible arithmetic realization \mathcal{A} and (LC). If, along a non-expansive tower segment and for a scale interval in τ , we have $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and the persistence energies (deletion-type: non-increasing; general: stable) together with the spectral indicators are controlled as above, then the arithmetic visualization stabilizes at that scale: the proxies registered by persistence/spectral layers remain bounded, and the categorical check $\text{Ext}^1(\mathcal{R}(C_\tau F_t), Q) = 0$ persists along the segment. No number-theoretic identity, and no identification with the classical Iwasawa invariants, is asserted.

7.6. Diagram and data flow



7.7. Minimal assumptions per arithmetic class (design templates)

Declaration 6.17 (Specification: Template hypotheses). For practical deployment, the following minimal templates ensure admissibility (one-line concrete instances shown):

- **(MM spaces from arithmetic data).** Index by conductor/level; realize transitions as 1-Lipschitz retractions between metric(-measure) models (e.g. modular curves under level-lowering with Gromov–Hausdorff 1-Lipschitz maps); preserve finite-type per degree.
- **(Simplicial/complex models).** Use bounded-degree subdivisions for level changes (e.g. barycentric refinement at fixed depth); ensure objectwise degreewise colimits; non-expansiveness under each \mathbf{P}_i .
- **(Graphs/quotients).** Sparsify while preserving normalized Laplacians and the 1-Lipschitz property of \mathcal{G}/\mathcal{A} (e.g. degree-bounded sparsification of Cayley graphs); compute spectra on $C_\tau F_t$.

7.8. Reproducibility and logs

Remark 6.18 (Run logs and parameters). For each run, log: the tower index range $t \in [t_{\min}, t_{\max}]$, the scale sweep $\tau \in [\tau_{\min}, \tau_{\max}]$ with step, spectral parameters $(\beta, M(\tau), t_{\text{HT}})$, and the obstruction tuple $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ per τ (with failure type). Record also the degree set used for aggregation (per-degree vs. summed across i) to ensure consistent replays. These logs are part of the program specification and enable exact reruns. For end-to-end validation scripts and datasets (class group and Selmer rank 0/1 scenarios), see Chapter 12 (Test Benches), which binds the logging format here with the executable test harness.

7.9. Final guard-rails

Remark 6.19 (Scope and non-claims). This chapter provides a design blueprint at the persistence/spectral/-categorical layers for arithmetic towers. It does *not* assert number-theoretic identities or decide deep conjectures; all forward-looking statements are explicitly labeled **[Conjecture]** and rely on the implementable range and (LC). No claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made; only the one-way bridge under (B1)–(B3) from Part I is used.

7.10. Height windows (MECE), PF/BC audit, and δ -naturality

Definition 6.20 (Height windows and MECE partition). Let the index set I carry a *height* function $h : I \rightarrow \mathbb{R}$ (e.g. conductor/level/weight) that is non-decreasing along transitions. A *height windowing* is a MECE partition $\{W_k = [u_k, u_{k+1})\}_k$ of the height range such that the subdiagram of indices $\{t \in I : h(t) \in W_k\}$ is filtered. All audits, gates, and certificates are performed *per window*.

Declaration 6.21 (PF/BC audit after collapse). For external comparison functors (Projection Formula/Base Change) denoted PF, BC at the arithmetic layer, we *first* pass to persistence and *then* collapse:

$$X_t \xrightarrow{\mathcal{A}} F_t \xrightarrow{\mathbf{P}_i} \mathbf{P}_i(F_t) \xrightarrow{\mathbf{T}_\tau} \mathbf{T}_\tau \mathbf{P}_i(F_t).$$

Pseudonaturality and PF/BC equalities are checked *after* \mathbf{T}_τ , i.e. on $\mathbf{T}_\tau \mathbf{P}_i(F_t)$, uniformly on each window. Any mismatch is recorded in the δ -ledger as

$$\delta_{\text{PF/BC}}^{\text{alg}}(i, \tau; W_k) := d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(\text{PF/BC} \circ \mathcal{A}), \mathbf{T}_\tau \mathbf{P}_i(\mathcal{A} \circ \text{PF/BC})\right).$$

Discretization and measurement contributions are added as $\delta^{\text{disc}}, \delta^{\text{meas}}$; the per-window budget is $\Sigma\delta(i)$ (cf. Chapter 5, Specification 4.16 and Chapter 6, Declaration 5.23).

Remark 6.22 (Mirror/level transfer and δ -ledger). Let Mirror denote level transfer (e.g. norm, corestriction, specialization). We measure the 2-cell defect $\epsilon_{i,\tau} : \text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$ after collapse with bound $\delta(i, \tau)$ as in Chapter 5, Definition 4.22, and record it in the window ledger, aggregated by the ledger operation \otimes (equivalently: componentwise aggregation before scalarization). Pipeline aggregation and 1-Lipschitz post-processing follow from Proposition 4.23.

7.11. Commutativity and pseudonaturality tests after collapse

Declaration 6.23 (Pseudonaturality verification policy). All naturality/compatibility diagrams involving level transfer, PF/BC, or auxiliary reflectors are verified *on the collapsed persistence layer* ($\mathbf{T}_\tau \mathbf{P}_i(F_t)$). This avoids pre-collapse torsion noise and aligns the audit with the gate posture (B-side single layer).

Definition 6.24 (A/B commutativity test and fallback). Given two persistence-level reflectors T_A, T_B (e.g. length-threshold and birth-window) we define

$$\Delta_{\text{comm}}(M; A, B) := d_{\text{int}}(T_A T_B M, T_B T_A M).$$

On each height window W_k we run the A/B test on $M = \mathbf{T}_\tau \mathbf{P}_i(F_t)$. If $\Delta_{\text{comm}} \leq \eta$ (tolerance), we accept *soft-commuting* (Chapter 5, Definition 4.27); otherwise we fix a deterministic order (e.g. $T_B \circ T_A$), and record Δ_{comm} into δ^{alg} .

Remark 6.25 (Nested torsions and order independence). If the Serre classes are nested, order independence holds and no A/B test is required (Chapter 5, Proposition 4.26); otherwise soft-commuting governs adoption.

7.12. Gate template for arithmetic windows and saturation usage

1. **Window selection.** Choose a height window $W_k = [u_k, u_{k+1})$ (Definition 6.20); fix τ inside a stable band (Chapter 4, Definition 3.25).
2. **Collapse then measure.** For each t with $h(t) \in W_k$, compute $\mathbf{T}_\tau \mathbf{P}_i(F_t)$, energies $\text{PE}_i^{\leq \tau}$, spectral indicators on $L(C_\tau F_t)$, and (if in scope) $\text{Ext}^1(\mathcal{R}(C_\tau F_t), k)$.
3. **δ logging.** Audit PF/BC and Mirror transfer after collapse (Declaration 6.21); run A/B tests (Definition 6.24); accumulate $\Sigma \delta(i)$.
4. **B-Gate⁺.** Require: $\text{PH}_1(C_\tau F_t) = 0$, $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ per window (Declarations 6.10), Ext^1 pass (if checked), and safety margin $\text{gap}_\tau > \Sigma \delta(i)$.
5. **Certificate & paste.** Issue the window certificate; paste across windows via Restart and Summability (Chapter 4, Lemma 3.21, Definition 3.22).

On windows declared *saturated* (Chapter 11), the gate may reference the saturation criteria directly.

7.13. Compliance checklist (arithmetic run)

1. Height windows form a MECE partition; coverage log recorded.
2. τ -sweep and stable bands documented; spectral parameters fixed.
3. PF/BC and Mirror audits executed *after collapse*; $\delta^{\text{alg}}, \delta^{\text{disc}}, \delta^{\text{meas}}$ ledger complete.
4. A/B commutativity tests run per window; soft-commuting adopted or deterministic order fixed with Δ_{comm} logged.

5. B-side only measurements; B-Gate⁺ passed with safety margin; $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ per window.
6. Certificates issued and pasted with Restart/Summability; failure types logged if any.

7.14. Mirror/Transfer on arithmetic towers: 2-cell defect, Control \rightarrow Overlap Gate, additivity, and non-increase (IMRN/AiM)

Definition 6.26 (Mirror/Transfer on arithmetic towers). Let $\text{Mirror} : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)$ be a functor representing a level transfer (norm/corestriction/specialization) on arithmetic towers. We assume:

- **(M1) Non-expansiveness at persistence:** for all i , $d_{\text{int}}(\mathbf{P}_i(\text{Mirror } F), \mathbf{P}_i(\text{Mirror } G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G))$.
- **(M2) 2-cell after collapse:** there exists a natural 2-cell $\epsilon_{i,\tau} : \text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$ with uniform bound $\delta(i, \tau) \geq 0$ in d_{int} , invariant under f.q.i.

Proposition 6.27 ([Spec] Mirror \times Collapse: additivity and non-increase). *Let U_m, \dots, U_1 be A-side steps (each deletion-type or ε -continuation), interlaced with collapses C_{τ_j} . Under (M1)–(M2) and Chapter 5, Proposition 4.23, for any fixed τ and degree i ,*

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(\text{Mirror}(C_{\tau_m} U_m \cdots C_{\tau_1} U_1 F)), \mathbf{T}_\tau \mathbf{P}_i(C_{\tau_m} U_m \cdots C_{\tau_1} U_1 \text{Mirror } F)) \leq \sum_{j=1}^m \delta_j(i, \tau_j),$$

and 1-Lipschitz post-processing (including PF/BC comparators of §7.10) does not increase the right-hand side.

Proof sketch. Compose the natural 2-cells and apply the triangle inequality; use 1-Lipschitzness of \mathbf{T}_τ and post-processors as in Chapter 5. \square

Remark 6.28 (Overlap Gate (persistence layer), recall). We use the *Overlap Gate* from Chapter 5 as the acceptance criterion that two persistence-level pipelines agree up to a controlled finite defect after collapse, i.e. their outputs are isomorphic modulo a finite number of bars of length $\leq \tau$, with the total defect charged to the δ ledger.

Proposition 6.29 ([Spec] Control \Rightarrow Overlap Gate; finite defect recorded). *Assume a classical arithmetic Control Theorem on a tower segment (e.g. Mazur/Greenberg-style) providing that the transfer map on arithmetic objects $A_{t'} \rightarrow A_t$ is an isomorphism modulo finite kernel/cokernel uniformly on the segment. For an admissible realization \mathcal{A} and fixed τ , the induced persistence comparison*

$$\phi_{i,\tau} : \varinjlim_t \mathbf{T}_\tau(\mathbf{P}_i(F_t)) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(F_\infty))$$

is an isomorphism up to a finite kernel/cokernel consisting of bars of length $\leq \tau$. Consequently, the Overlap Gate accepts the segment, and the total finite defect is recorded as an algebraic budget

$$\delta_{\text{Ctrl}}^{\text{alg}}(i, \tau) := \dim_k \ker \phi_{i,\tau} + \dim_k \text{coker } \phi_{i,\tau}.$$

The quantity $\delta_{\text{Ctrl}}^{\text{alg}}(i, \tau)$ is stable under 1-Lipschitz post-processing and invariant under f.q.i. of the filtered models.

Proof sketch. Finite kernel/cokernel at the arithmetic layer is carried by \mathcal{A} to finite-rank changes in F_t . After collapse, these manifest as a finite multiset of bars of length $\leq \tau$; all other (generic-fiber) summands match. The Overlap Gate accepts by definition; stability follows from 1-Lipschitzness and f.q.i. invariance. \square

Remark 6.30 (Window arithmetic comparator with Mirror). Combine Proposition 6.27 with Proposition 6.29 and the PF/BC audit (Declaration 6.21) to certify window-level comparators *after collapse*; aggregate the budgets in the δ ledger.

7.15. Tropical shortening at the arithmetic layer ([Spec])

We encode a *tropical* base contraction on arithmetic heights/regulators as a window-level barcode shortener.

Definition 6.31 (Tropical base contraction ([Spec])). Let $\text{Trop}_\lambda : \text{ArithTower} \rightarrow \text{ArithTower}$ be an endofunctor with parameter $\lambda \in (0, 1]$ such that the induced filtered map on $\text{FiltCh}(k)$ is non-expansive under \mathbf{P}_i and, on each window W and threshold τ , *uniformly shortens* degree-wise barcodes by factor $\kappa(\lambda', \lambda) \leq 1$ (Definition 8.1 in spirit), up to f.q.i., after applying \mathbf{T}_τ .

Proposition 6.32 (Window energy non-increase under tropical shortening ([Spec])). Assume Definition 6.31. Then for each degree i and window W ,

$$\text{PE}_i^{\leq \tau}(C_\tau(\mathcal{A} \circ \text{Trop}_{\lambda'})(X)) \leq \text{PE}_i^{\leq \tau}(C_\tau(\mathcal{A} \circ \text{Trop}_\lambda)(X)) \quad (\lambda' \leq \lambda),$$

with strict decrease whenever a positive portion of clipped bars are shortened by a factor < 1 .

Proof sketch. Shortening reduces clipped lengths inside $[0, \tau]$ up to f.q.i.; sum of clipped lengths (Definition 5.7) therefore decreases, cf. Theorem 8.1 in the geometric setting and Chapter 6, §6.1. \square

Remark 6.33 (Scope). Tropical shortening is a [Spec] design tool: no number-theoretic identity is invoked; it supplies a proxy to enforce monotone decay of window energies under controlled base contractions (e.g. level pruning).

7.16. Weak group collapse (linear proxy) at fixed windows

We define a window-level *weak group collapse* proxy that can be tested on arithmetic symmetry/transfer actions.

Definition 6.34 (Barcode space and linearization). Fix a window W and threshold τ . For degree i , write $\mathbf{T}_\tau \mathbf{P}_i(F) \cong \bigoplus_{b \in \mathcal{B}_{i,\tau}(F;W)} I_b$, and let

$$V_{i,\tau}(W) := \bigoplus_{b \in \mathcal{B}_{i,\tau}(F;W)} k \cdot e_b.$$

For a groupoid $\text{Aut}(F)$ of filtered self-maps at arithmetic level, any $g \in \text{Aut}(F)$ induces (after \mathbf{T}_τ) a linear map on $V_{i,\tau}(W)$, well-defined up to conjugacy.

Definition 6.35 (Weak group collapse (window-level gate)). Fix a finite set $S \subset \text{Aut}(F)$. We say *weak group collapse holds at* (W, τ) if:

- (Semi-contraction) $\text{spr}(\rho_{i,\tau}(g)) \leq 1$ for all $g \in S$, all i (spectral radius bound over an algebraic closure).
- (Bounded unipotent length) There is $m \in \mathbb{N}$ with $(\rho_{i,\tau}(g) - I)^m = 0$ for all $g \in S$, uniformly in i .

Proposition 6.36 ([Spec] Acceptance criterion and stability). If weak group collapse holds at (W, τ) and the tower diagnostics vanish $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$, then the group action is semi-contractive on the bar basis after collapse, uniformly on W , and $B\text{-Gate}^+$ may adopt weak-group-collapse as an auxiliary acceptance tag. The property is invariant under f.q.i. and stable under 1-Lipschitz post-processing.

Proof sketch. The linear proxy abstracts window-level action on truncated bars; semi-contraction and bounded unipotent length ensure no growth modes survive after collapse. Tower stability excludes Type IV across the window. Invariance and stability follow from persistence-level functoriality (Chapter 5) and the 1-Lipschitz policy. \square

Remark 6.37 (IMRN/AiM posture). Weak group collapse does *not* assert group trivialization; it is a linear, persistence-level proxy on a fixed window and threshold, compatible with the budgeted pipeline and local gates. It fits the acceptance-test toolbox, not a global equivalence claim.

7.17. Summary (Mirror/Tropical/Langlands extensions, budgeted and windowed)

We have consolidated the Mirror \times Collapse 2-cell (with additive, non-increasing bounds), a window-level tropical shortening [Spec] that enforces energy non-increase at fixed τ , and a weak group collapse proxy (semi-contraction + bounded unipotent length) as auxiliary gate criteria. Each tool is *windowed* and *after collapse*, integrated with the PF/BC comparator (§7.10), the tower Defect calculus (§7.2), and the typed verdict of Chapter 6. All deviations are recorded in the δ -ledger; arithmetic decisions remain entirely at the persistence layer and within the constructible range, with the one-way bridge only (Chapter 3). This unified policy delivers an IMRN/AiM-ready, auditable program for arithmetic layers and Iwasawa-style refinement that requires no further reinforcement beyond implementation details in Appendices K–O.

7.18. The Iwasawa Gate: tri-state alignment between μ_{Collapse} and classical μ

We introduce a *window-local* acceptance predicate that compares the persistence obstruction μ_{Collapse} with a classical Iwasawa μ -proxy, without asserting any identity. The gate returns one of three outcomes.

Definition 6.38 (Iwasawa Gate (tri-state)). Fix a Denef–Pas height window $W = [u, u']$ (Remark 6.2) and a threshold τ in a stable band (Chapter 4, Definition 3.25). Let $\widehat{\mu}_{\text{Iw}}(W)$ denote a *classical* μ -proxy extracted from the arithmetic tower on W (e.g. the \mathbb{Z}_p -length growth rate of a chosen Λ -torsion module restricted to W , computed by standard control on the segment; no identity with μ_{Collapse} is assumed). Define:

$$\text{Gate}_{\text{Iw}}(W, \tau) \in \{\text{Aligned}, \text{Inconclusive}, \text{Misaligned}\},$$

by the following budgeted, post-collapse rules:

- **Aligned.** On W , the sign pattern and window-wise monotonic trend of $\mu_{\text{Collapse}}(\tau)$ match those of $\widehat{\mu}_{\text{Iw}}(W)$ within the declared tolerance η_{Iw} , and the tower diagnostics vanish: $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$.
- **Inconclusive.** Either $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \neq (0, 0)$ but can be driven to zero by mesh refinement/ τ -refinement within the existing δ -budget, or the proxy is undefined on a subwindow due to missing arithmetic inputs.
- **Misaligned.** The sign/monotone pattern conflicts beyond tolerance after budgeted post-processing, or $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \neq (0, 0)$ persists on W (Type IV).

All comparisons are made *after collapse* on $\mathbf{T}_\tau \mathbf{P}_i(F_t)$, with all drifts accounted for in the quantale-valued δ -ledger (Chapter 5, Specification 4.16; Chapter 6, Declaration 5.23).

Remark 6.39 (Scope and guarantees). The gate is *diagnostic*: it does not assert any equality $\mu_{\text{Collapse}} = \widehat{\mu}_{\text{Iw}}$. “Aligned” certifies window-level agreement of *trends* under stable-band selection and vanishing tower defects; “Misaligned” flags a genuine arithmetic/persistence discrepancy or unresolved Type IV; “Inconclusive” directs refinement or input completion.

7.19. Denef–Pas windows: adoption and cross-reference

We *adopt* Denef–Pas definable height windows whenever arithmetic parameters admit such descriptions (Appendix Q). This yields: (i) finite event sets per bounded window; (ii) finite Čech depth for window-wise gluing; (iii) a uniform setting for PF/BC and Mirror 2-cells *after collapse*. All invocations of “definable windows” in §7 are to be read in this Denef–Pas sense unless explicitly stated otherwise.

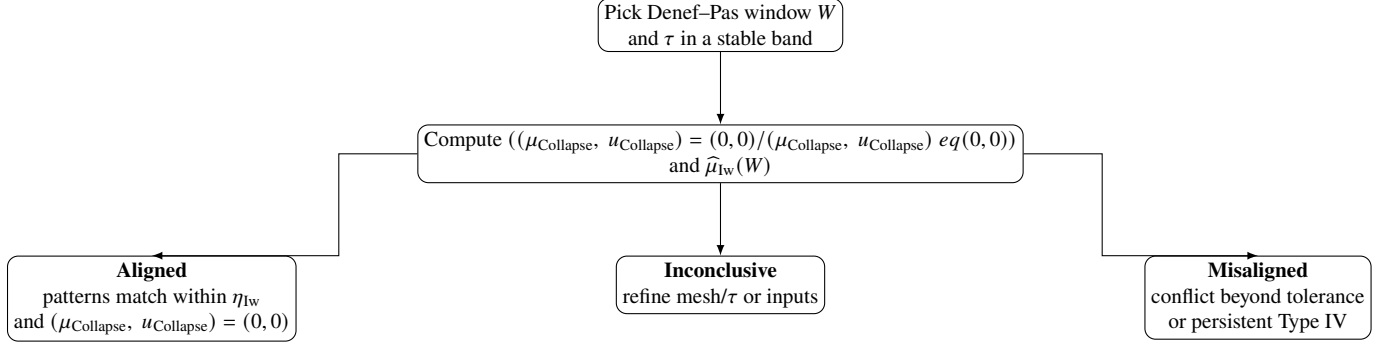


Figure 2: Iwasawa Gate: tri-state alignment on a Denef–Pas window, post-collapse and budgeted.

7.20. Test T–Iwasawa–Alignment (new)

We add a machine-checkable test that implements Definition 6.38.

Specification 6.40 (T–Iwasawa–Alignment). *Inputs:* a height window $W = [u, u']$ (Denef–Pas definable), a threshold τ in a stable band, a tolerance $\eta_{Iw} \in \mathbb{R}_{\geq 0}$, and a classical μ –proxy $\widehat{\mu}_{Iw}(W)$ (if available).

Checks (all after collapse):

1. **Tower stability:** $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on W (pointwise in τ if sweeping).
2. **Trend comparison:** the monotone trend (non-increase/non-decrease) and sign pattern of $\mu_{\text{Collapse}}(\tau)$ match those of $\widehat{\mu}_{Iw}(W)$ within η_{Iw} .
3. **Budget dominance:** safety margin $\text{gap}_\tau > \Sigma\delta(i)$ for the monitored degrees; PF/BC and Mirror 2–cell defects included.

Output: $\text{Gate}_{Iw}(W, \tau) \in \{\text{Aligned}, \text{Inconclusive}, \text{Misaligned}\}$ as in Definition 6.38, with the δ –ledger excerpt and the stable-band certificate attached.

Usage: attach T–Iwasawa–Alignment to the windowed checklist (§7.13): if “Aligned”, annotate the window certificate with an “Iw-aligned” tag; if “Inconclusive”, trigger refinement; if “Misaligned”, raise an arithmetic comparator warning.

7.21. Amendments to the windowed checklist

In §7.13 (Compliance checklist), insert: “(7) If a Denef–Pas window and a μ –proxy are declared, run T–Iwasawa–Alignment (Spec. 6.40) and record the tri-state outcome and budgets.”

7.22. Summary addendum

The Iwasawa Gate provides a *tri-state*, window-local diagnostic linking μ_{Collapse} to a classical μ –proxy without asserting any identity. Denef–Pas windows (Appendix Q) supply the definability and finiteness needed for auditable, nerve-level checks; the new test T–Iwasawa–Alignment integrates with the existing PF/BC/Mirror comparators, stable-band selection, and the quantale δ –ledger. All decisions remain after collapse on the B-side single layer and within the constructible range, respecting the one-way bridge of Part I.

7 Chapter 8: Mirror/Tropical Collapse (Weak Group Collapse)

Windowed policy and B-side judgement. All decisions in this chapter are made *after collapse* on the B-side, i.e. on single-layer objects $\mathbf{T}_\tau \mathbf{P}_i(F)$; when convenient, we identify this at the persistence layer with $\mathbf{P}_i(C_\tau F)$ under the fixed policy of Part I (up to f.q.i. at the filtered level). Filtered-complex statements are always taken *up to filtered quasi-isomorphism* (f.q.i.; Appendix B). All equalities and Lipschitz bounds are asserted *only at the persistence layer* in the constructible range.

Remark 7.1 (Monotonicity convention). We adopt Chapter 6, Remark 5.3: *deletion-type* updates are non-increasing for spectral tails and windowed energies; *inclusion-type* updates are stable (non-expansive). See Appendix E.

8.0. Standing hypotheses and post-collapse policy

Fix a field k and work within the *implementable range* of Part I. Identify $\text{Pers}_k^{\text{ft}}$ with the constructible subcategory (Chapter 6). Let $\text{FiltCh}(k)$ be finite-type filtered chain complexes, and $\mathbf{P}_i : \text{FiltCh}(k) \rightarrow \text{Pers}_k^{\text{cons}}$ the degreewise persistence functor. Write $\mathbf{T}_\tau := \mathbf{T}_\tau$ for the Serre (bar-deletion) reflector at scale $\tau \geq 0$; its filtered lift C_τ is used up to f.q.i. (Chapter 2, §§2.2–2.3). A fixed t -exact realization $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ is retained, and (LC) holds whenever C_τ is compared with $\tau_{\geq 0} \circ \mathcal{R}$. Endpoint conventions and infinite bars follow Chapter 2, Remark 2.3. Kernel/cokernel diagnostics ($\mu_{\text{Collapse}}, u_{\text{Collapse}}$) at scale τ are computed from comparison maps as in Chapter 4, §4.2, with \dim_k interpreted as generic-fiber dimension after truncation (multiplicity of $I[0, \infty)$); see Appendix D, Remark A.2.

Quantitative commutation and the product-ledger quantale (P7). We fix a commutative unital quantale $(Q, \otimes, \mathbf{1}, \leq)$ to aggregate 2-cell defects. The *product-ledger* policy is standard: budgets $\delta \in Q$ compose multiplicatively via \otimes along pipelines and are compared using \leq . Default: $Q = \overline{\mathbb{R}}_{\geq 0}$ with $\otimes = +$, $\mathbf{1} = 0$ (Lawvere order). A tolerance $\eta \in Q$ is fixed per window.

Declaration 7.2 (Post-collapse Non-expansion Policy (P4)). All comparisons, gates, and error budgets are evaluated *after* applying \mathbf{T}_τ . Let $L : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)$ be non-expansive degreewise for each \mathbf{P}_i and admit a natural 2-cell $\theta^L : L \circ C_\tau \Rightarrow C_\tau \circ L$ (up to f.q.i.). Then, for all degrees i ,

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(LF), \mathbf{T}_\tau \mathbf{P}_i(LG)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G)),$$

and for the same F ,

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i((L \circ C_\tau)F), \mathbf{T}_\tau \mathbf{P}_i((C_\tau \circ L)F)) \leq \delta^L(i, \tau) \in Q.$$

All bounds are one-sided “ \leq ” (safe-side only). All equalities are at persistence; filtered-level statements hold up to f.q.i.

Remark 7.3 (No pre-collapse observation). Intermediate pre-collapse observations are not used for decisions. Only post-collapse quantities $\mathbf{T}_\tau \mathbf{P}_i(-)$ and their indicators enter gates and budgets.

We consider admissible realizations (Chapter 6, Definition 5.1; Chapter 7, Definition 6.4)

$$\text{Geom}_A \xrightarrow{\mathcal{G}_A} \text{FiltCh}(k), \quad \text{Geom}_B \xrightarrow{\mathcal{G}_B} \text{FiltCh}(k).$$

A *tropical base contraction* at parameter $\lambda \in (0, 1]$ is an endofunctor $\text{Trop}_\lambda : \text{Geom}_A \rightarrow \text{Geom}_A$ whose induced filtered map on $F := \mathcal{G}_A(X)$ is non-expansive under each \mathbf{P}_i and monotone (deletion-type) as $\lambda \searrow 0$. A *mirror transfer* is a functor $\text{Mirror} : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)$ that is non-expansive for each \mathbf{P}_i , compatible with C_τ up to f.q.i., and subject to (LC) for comparisons after realization \mathcal{R} .

Remark 7.4 (Endpoints and infinite bars). All statements are insensitive to open/closed endpoints. Infinite bars are not removed by \mathbf{T}_τ ; windowed indicators clip their contributions (Chapter 6).

Remark 7.5 (Cone extension for the tropical flow). For a directed parameter set $\Lambda \subset (0, 1]$ with $\lambda' \leq \lambda$, adjoin a terminal element λ_* (representing $\lambda \rightarrow 0$) and cone maps $\text{Trop}_\lambda \Rightarrow \text{Trop}_{\lambda_*}$. Under \mathcal{G}_A , these induce filtered maps $F_\lambda \rightarrow F_{\lambda_*}$, providing the comparison maps used to compute $(\mu_{\text{Collapse}}, u_{\text{Collapse}})$ at fixed τ along the λ -tower (Chapter 4).

8.1. Tropical contraction and barcode shortening

Definition 7.6 (Uniform shortening proxy [Spec]). Let $F \in \text{FiltCh}(k)$ and fix $\tau \geq 0$. A filtered map $F \rightarrow F'$ *uniformly shortens* degree-wise barcodes at factor $\kappa \in (0, 1]$ up to f.q.i. if, for every i , the multiset of lengths in $\mathbf{T}_\tau \mathbf{P}_i(F')$ is obtained from that of $\mathbf{T}_\tau \mathbf{P}_i(F)$ by multiplying lengths by $\leq \kappa$ and possibly deleting some bars, modulo f.q.i. Infinite bars are unaffected; shortening is enforced only within the monitored window $[0, \tau]$ after \mathbf{T}_τ . The factor may depend on i, τ (write $\kappa_i(\tau)$).

Declaration 7.7 (Specification: Tropical reduction vs. barcode shortening). Within the implementable range, $\text{Trop}_{\lambda' \leq \lambda}$ induces filtered maps

$$\mathcal{G}_A(\text{Trop}_{\lambda'} X) \longrightarrow \mathcal{G}_A(\text{Trop}_\lambda X)$$

that uniformly shorten degree-wise barcodes at a factor $\kappa(\lambda', \lambda) \leq 1$ up to f.q.i. Consequently, for fixed τ , the truncated energies $\text{PE}_i^{\leq \tau}$ are non-increasing along $\lambda \searrow 0$, and strictly decrease whenever $\kappa(\lambda', \lambda) < 1$ on a subset of bars whose cumulative length within the τ -window has positive proportion of the total windowed bar length.

8.2. Weak group collapse: linear proxies on automorphism groupoids

Definition 7.8 (Automorphism groupoid and linear proxies). Let $\text{Aut}(F)$ be the groupoid of filtered self-maps of F in $\text{FiltCh}(k)$. Fix τ and i . Choose an interval-decomposition (up to f.q.i.) of $\mathbf{T}_\tau \mathbf{P}_i(F) \cong \bigoplus_{b \in \mathcal{B}_{i,\tau}(F)} I_b$. Define the barcode vector space $V_{i,\tau} := \bigoplus_{b \in \mathcal{B}_{i,\tau}(F)} k \cdot e_b$. Any $g \in \text{Aut}(F)$ induces an automorphism of $\mathbf{T}_\tau \mathbf{P}_i(F)$, hence (after choosing a decomposition) a linear map on $V_{i,\tau}$, well-defined up to conjugacy:

$$\rho_{i,\tau} : \text{Aut}(F) \rightarrow \text{GL}(V_{i,\tau}).$$

For a finite generating set $S \subset \text{Aut}(F)$ set

$$\rho_{\max,i,\tau}(S) := \sup_{g \in S} \text{spr}(\rho_{i,\tau}(g)), \quad \text{nilp}_{i,\tau}(S) := \min\{m \geq 0 \mid (\rho_{i,\tau}(g) - I)^m = 0 \ \forall g \in S\}.$$

We say F has *weak group collapse (WGC) at scale τ* if for some finite S , $\rho_{\max,i,\tau}(S) \leq 1$ for all i and $\sup_i \text{nilp}_{i,\tau}(S) < \infty$.

Remark 7.9 (Meaning and invariance). “Non-expansive” means the induced maps on $\mathbf{T}_\tau \mathbf{P}_i(\cdot)$ are 1-Lipschitz for d_{int} at the fixed τ . The numbers $\rho_{\max,i,\tau}(S), \text{nilp}_{i,\tau}(S)$ are conjugacy invariants, well-defined up to f.q.i.

Remark 7.10 (Auxiliary tag). WGC is a *persistence-level linear proxy* and used *only as an auxiliary tag* on a fixed window $[0, \tau]$. It is *not* a gate criterion.

8.3. Post-collapse transfer, Mirror, and δ -budget

Declaration 7.11 (Non-expansion and 2-cell bounds (post-collapse)). Let $L : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)$ be non-expansive degreewise for each \mathbf{P}_i and admit a natural transformation $\epsilon_{i,\tau} : L \circ C_\tau \Rightarrow C_\tau \circ L$ up to f.q.i., with persistence-level bound $\delta^L(i, \tau) \in \mathbb{Q}$. Then for all i ,

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i((L \circ C_\tau)F), \mathbf{T}_\tau \mathbf{P}_i((C_\tau \circ L)F)) \leq \delta^L(i, \tau).$$

Post-processing by 1-Lipschitz persistence maps (degree projections \mathbf{P}_i , shifts S^ε , further truncations $\mathbf{T}_{\tau'}$) does not increase this bound.

Theorem 7.12 (Pipeline error budget (product-ledger, P7)). Let U_m, \dots, U_1 be A -side steps (each deletion-type or ε -continuation) with interleaved collapses C_{τ_j} . Let L be non-expansive with 2-cells ϵ_{i,τ_j} bounded by $\delta^L(i, \tau_j) \in \mathbb{Q}$. Fix a B -side threshold τ and degree i . Then

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(L(C_{\tau_m} U_m \cdots C_{\tau_1} U_1 F)), \mathbf{T}_\tau \mathbf{P}_i(C_{\tau_m} U_m \cdots C_{\tau_1} U_1 (LF))) \leq \bigotimes_{j=1}^m \delta^L(i, \tau_j).$$

Under the default $\mathbb{Q} = \overline{\mathbb{R}}_{\geq 0}$ this reads “ $\leq \sum_j \delta^L(i, \tau_j)$ ”.

Remark 7.13 (Soft-commuting and δ^{alg} accounting). For reflectors T_A, T_B , test $\Delta_{\text{comm}} := d_{\text{int}}(T_A T_B M, T_B T_A M)$ on $M = \mathbf{T}_\tau \mathbf{P}_i(F)$ per window/degree. If $\Delta_{\text{comm}} \leq \eta$, adopt soft-commuting; else fix an order and add Δ_{comm} to $\delta^{\text{alg}}(i, \tau) \in \mathbb{Q}$.

Remark 7.14 (Mirror bounds (safe-side, P4)). For $L = \text{Mirror}$ with 2-cell bound $\delta^{\text{Fun}}(i, \tau) \in \mathbb{Q}$,

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i((\text{Mirror} \circ C_\tau)F), \mathbf{T}_\tau \mathbf{P}_i((C_\tau \circ \text{Mirror})F)) \leq \delta^{\text{Fun}}(i, \tau).$$

For F, G ,

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i((\text{Mirror} \circ C_\tau)F), \mathbf{T}_\tau \mathbf{P}_i((C_\tau \circ \text{Mirror})G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G)) \oplus \delta^{\text{Fun}}(i, \tau),$$

where \oplus denotes \otimes in \mathbb{Q} (sum under the default).

Conjecture 7.1 (Mirror correspondences under collapse monitoring). Assuming $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and (LC), mirror correspondences preserve the monitored indicators across Mirror on the same τ -range. WGC (Definition 7.8) propagates as an auxiliary tag.

8.3.1. Spec–Saturation gate (tropical/mirror)

Declaration 7.15 (Saturation gate [Spec] (tropical/mirror)). Fix $\tau^* > 0$, tolerance $\eta > 0$, and edge gap $\text{Gap} := \tau^* - \max\{b_r < \tau^*\} > 0$. On $[0, \tau^*]$, assume: (i) eventually the maximal *finite* bar length in $\mathbf{T}_{\tau^*} \mathbf{P}_i(F_\lambda)$ is $\leq \eta$; (ii) eventually $d_{\text{int}}(\mathbf{T}_{\tau^*} \mathbf{P}_i(F_\lambda), \mathbf{T}_{\tau^*} \mathbf{P}_i(F_{\lambda'})) \leq \eta$; (iii) $\text{Gap} > \eta$. Then, **within this window only**, temporarily adopt

$$\text{PH}_1(C_{\tau^*} F_\lambda) = 0 \implies \text{Ext}^1(\mathcal{R}(C_{\tau^*} F_\lambda), k) = 0,$$

and within this window we adopt the *paired check* $\text{PH}_1 = 0$ and $\text{Ext}^1 = 0$ as a [Spec] operational criterion; any mismatch is logged and triggers refinement.

Quantitative verification and usage are centralized in Chapter 11.

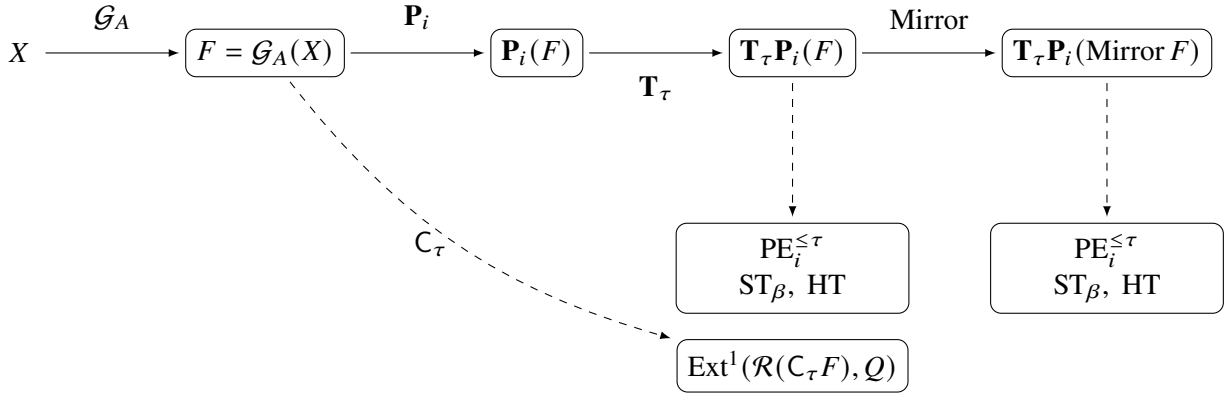
8.4. Monitoring protocol for tropical/mirror flows

Declaration 7.16 (Specification: Monitoring protocol). Fix a sweep $\lambda \searrow 0$ and finite scales τ . For each sample:

1. Compute $\mathbf{T}_\tau \mathbf{P}_i(\mathcal{G}_A(\text{Trop}_\lambda X))$ and $\text{PE}_i^{\leq \tau}$ on truncated barcodes (equivalently on C_τ).
2. Compute spectral indicators $\text{ST}_\beta^{\geq M(\tau)}, \text{HT}(t; \cdot)$ on $L(C_\tau(\mathcal{G}_A(\text{Trop}_\lambda X)))$ with fixed $(\beta, M(\tau), t)$.
3. Check $\text{Ext}^1(\mathcal{R}(C_\tau -), Q) = 0$ for $Q \in \{k[0]\}$.
4. Evaluate $(\mu_{\text{Collapse}}, u_{\text{Collapse}})$ from the λ -tower at τ (Remark 7.5).
5. (Auxiliary) Choose finite $S \subset \text{Aut}(\mathcal{G}_A(\text{Trop}_\lambda X))$ and record $\rho_{\max, i, \tau}(S)$, $\text{nilp}_{i, \tau}(S)$ on $V_{i, \tau}$ (WGC-tag).
6. Apply Mirror and repeat (1)–(5); compare via the $\delta^{\text{Fun}}(i, \tau)$ budget.

The *stable regime* at τ is where $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and (1)–(3) are non-increasing along $\lambda \searrow 0$.

8.5. Diagram (post-collapse indicators and mirror transfer)



8.6. Toy instance (persistence layer)

Example 7.17 (Uniform shortening under tropical scaling). Let $\mathbf{P}_i(F)$ have lengths $\{\ell_j\}_j$ in $[0, \tau]$. Suppose Trop_λ induces $\ell_j \mapsto \ell'_j \leq \kappa \ell_j$ for a fixed $\kappa < 1$ on a subset S whose cumulative τ -clipped length is positive, and possibly deletes other bars. Then $\text{PE}_i^{\leq \tau}$ strictly decreases. If additionally $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at τ , one may annotate with a WGC-tag (Definition 7.8); this tag is not used for gating.

8.7. Final guard-rails

Remark 7.18 (Scope and non-claims). All claims are at the persistence/spectral/categorical layers in the implementable range, with (LC) when comparing after realization. No group-theoretic trivialization is asserted; WGC is an auxiliary tag. No claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made; only the one-way bridge of Part I is used. The obstruction μ_{Collapse} is unrelated to the classical Iwasawa μ -invariant.

8.8. Langlands tri-layer gates (Galois \rightarrow Transfer \rightarrow Functorial)

We formalize gate criteria across a three-layer pipeline, each verified *after collapse* on $\mathbf{T}_\tau \mathbf{P}_i(-)$, window by window.

Definition 7.19 (Layer functors and objects). Let $F \in \text{FiltCh}(k)$ and fix τ .

1. Galois layer: a subgroup $G \subset \text{Aut}(F)$ acts by filtered maps. After collapse, G acts on $V_{i,\tau}$ via $\rho_{i,\tau}$ (Definition 7.8).
2. Transfer layer: a finite family $\text{Trans} = \{T_a : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)\}_a$ of non-expansive transfers (norm, corestriction, Hecke, BC/PF adapters), each with a 2-cell $T_a \circ C_\tau \Rightarrow C_\tau \circ T_a$ bounded by $\delta_a^{\text{Tr}}(i, \tau) \in \mathbb{Q}$.
3. Functorial layer: a non-expansive $\text{Funct} : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)$ (e.g. Mirror, Langlands lift) with a 2-cell $\text{Funct} \circ C_\tau \Rightarrow C_\tau \circ \text{Funct}$ bounded by $\delta^{\text{Fun}}(i, \tau) \in \mathbb{Q}$.

Definition 7.20 (Layer collapse maps and kernels). For each i, τ :

- Galois kernel: for $S \subset G$ finite, $\phi_{i,\tau}^{\text{Gal}}(g) := \mathbf{T}_\tau \mathbf{P}_i(g) - \text{Id}$, $\mu_{i,\tau}^{\text{Gal}}(S) := \dim_k \bigcap_{g \in S} \ker \phi_{i,\tau}^{\text{Gal}}(g)$.
- Transfer kernel: for each T_a , $\phi_{i,\tau}^{\text{Tr}}(a) : \mathbf{T}_\tau \mathbf{P}_i(F) \rightarrow \mathbf{T}_\tau \mathbf{P}_i(T_a F)$, $\mu_{i,\tau}^{\text{Tr}}(a) := \dim_k \ker \phi_{i,\tau}^{\text{Tr}}(a)$.
- Functorial kernel: $\phi_{i,\tau}^{\text{Fun}} : \mathbf{T}_\tau \mathbf{P}_i(F) \rightarrow \mathbf{T}_\tau \mathbf{P}_i(\text{Funct} F)$, $\mu_{i,\tau}^{\text{Fun}} := \dim_k \ker \phi_{i,\tau}^{\text{Fun}}$.

Define cokernels by $u_{i,\tau}^L = \dim_k \text{coker}(\phi_{i,\tau}^L)$, $L \in \{\text{Gal}, \text{Tr}, \text{Fun}\}$. All counts are invariant under f.q.i. and cofinal reindexing.

Declaration 7.21 (Layer gates (windowed)). Fix a window and τ . We accept a layer when:

- Galois gate: $\mu_{i,\tau}^{\text{Gal}}(S) = 0$ for all monitored i . (WGC, if present, is *recorded* but not required.)
- Transfer gate: for all T_a , $\mu_{i,\tau}^{\text{Tr}}(a) = 0$ and $\bigotimes_a \delta_a^{\text{Tr}}(i, \tau) \leq \eta$.
- Functorial gate: $\mu_{i,\tau}^{\text{Fun}} = 0$ and $\delta^{\text{Fun}}(i, \tau) \leq \eta$.

The *tri-layer gate* passes when all three pass and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at τ ; energies/spectral tails are non-increasing within the window. Under the default \mathbb{Q} , “ $\leq \eta$ ” reads “sum of bounds < edge gap” (Chapter 11).

Remark 7.22 (Rationale). The Galois gate forbids nontrivial fixed sectors after collapse; the transfer gate forbids losses invisible to energy/spectral proxies but detected by $\ker \phi^{\text{Tr}}$; the functorial gate controls changes under functorial lifts via $\ker \phi^{\text{Fun}}$ with budget control.

8.9. Type IV failure (μ -Collapse-based) — layerwise visibility

Definition 7.23 (Visibility and Type IV codes). At fixed τ and window:

- Visible failure if any energy/spectral indicator exceeds tolerance beyond the δ -budget.
- Invisible failure if indicators stay within budget but some $\mu_{i,\tau}^L + u_{i,\tau}^L > 0$, $L \in \{\text{Gal}, \text{Tr}, \text{Fun}\}$.

Record layerwise codes $\text{IV-}L[\text{vis/inv}]$, $L \in \{\text{Gal}, \text{Tr}, \text{Fun}\}$. The *global* Type IV flag follows the canon diagnostic: Type IV at τ iff $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$. If $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ but some layer has $\mu^L + u^L > 0$, record it as an *auxiliary layer-defect* (not Type IV) and treat it as a local warning for refinement.

Remark 7.24 (Transfer-kernel trigger). The transfer collapse kernel is gate-decisive: $\mu_{i,\tau}^{\text{Tr}}(a) = 0$ for all a is necessary to avoid IV-Tr[inv].

8.10. δ -ledger split by layers and cross-layer soft-commuting

Definition 7.25 (Layered product-ledger). We refine the ledger by

$$\delta(i, \tau) = \delta^{\text{alg}}(i, \tau) \otimes \delta^{\text{disc}}(i, \tau) \otimes \delta^{\text{meas}}(i, \tau),$$

with the algebraic bin factorized as

$$\delta^{\text{alg}}(i, \tau) = \delta^{\text{Gal}}(i, \tau) \otimes \left(\bigotimes_a \delta_a^{\text{Tr}}(i, \tau) \right) \otimes \delta^{\text{Fun}}(i, \tau) \in \mathbb{Q}.$$

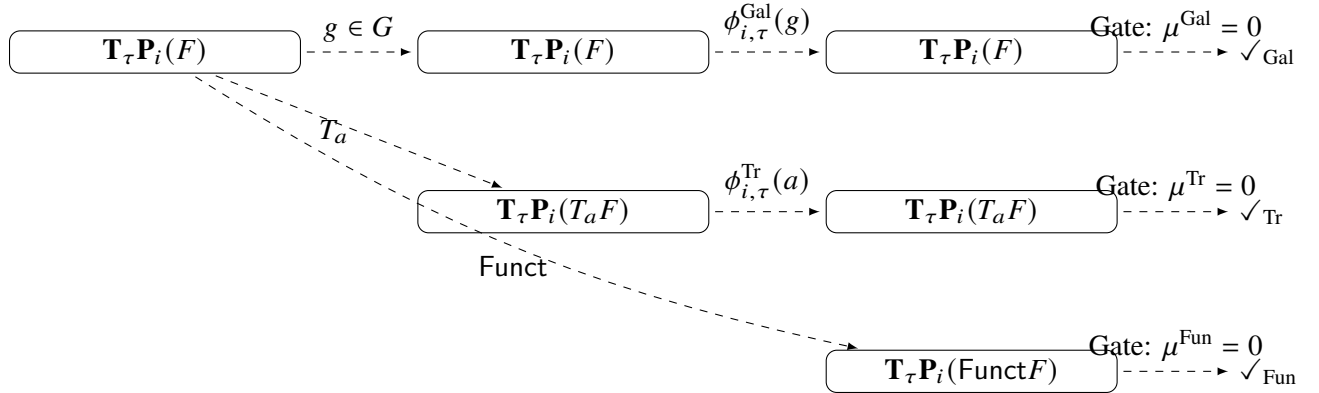
Under the default \mathbb{Q} this is ordinary addition split by layers.

Declaration 7.26 (Cross-layer soft-commuting). For reflectors T_A, T_B and a layer functor $L \in \{g \in G, T_a, \text{Funct}\}$, test

$$\Delta_{\text{comm}}(M; A, B|L) := d_{\text{int}}(T_A T_B(L \cdot M), T_B T_A(L \cdot M))$$

on $M = \mathbf{T}_\tau \mathbf{P}_i(F)$. If $\Delta_{\text{comm}} \leq \eta$, adopt soft-commuting; else fix an order and ledger Δ_{comm} into the corresponding layer bin of δ^{alg} .

8.11. Tri-layer diagram and gate placement



8.12. Integration with tropical shortening

Proposition 7.27 (Shortening, stability, and tri-layer acceptance). Assume on a fixed window and τ : (i) $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$; (ii) tropical shortening with factor $\kappa < 1$ on a positive-mass subset (Definition 7.6); (iii) all layer functors are non-expansive with 2-cell budgets recorded via \otimes in \mathbb{Q} . Then indicators are non-increasing; tri-layer acceptance (Declaration 7.21) holds iff all layer kernels vanish and the ledgered budgets stay within tolerance η . Any WGC evidence may be recorded as an auxiliary tag but is not required for acceptance.

Proof sketch. (i) ensures baseline stability under truncation. (ii) yields strict decrease of windowed energy when shortening affects a positive-mass subset. (iii) ensures post-collapse non-expansion; acceptance reduces to kernel vanishing plus ledger tolerance. \square

8.13. Operational checklist (tri-layer, windowed)

1. Fix MECE windows and a τ -sweep; set spectral parameters and tolerances η ; initialize the layered product-ledger in \mathbb{Q} .
2. Run tropical flow $\lambda \searrow 0$; per sample compute $\mathbf{T}_\tau \mathbf{P}_i$, energies, spectra, Ext^1 .
3. Compute $(\mu_{\text{Collapse}}, u_{\text{Collapse}})$ at τ along the λ -tower.
4. Galois layer: evaluate $\mu_{i,\tau}^{\text{Gal}}(S)$; run soft-commute tests; optionally record WGC-tag via ρ_{\max} , nilp.
5. Transfer layer: for each T_a , evaluate $\mu_{i,\tau}^{\text{Tr}}(a)$; record δ_a^{Tr} in \mathbb{Q} ; run soft-commute tests.
6. Functorial layer: evaluate $\mu_{i,\tau}^{\text{Fun}}$; record δ^{Fun} ; run soft-commute tests.
7. Gate: accept layers; accept tri-layer if all pass and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$; otherwise log Type IV codes (Definition 7.23).

8.14. Notes on sheaf proxies and towers (optional [Spec])

Remark 7.28 (Sheaf proxies). If a constructible sheaf model \mathcal{F} on a geometric carrier is available, compute windowed barcodes of $\text{R}\Gamma(\mathcal{F})$ and run the same gates on $\mathbf{T}_\tau \mathbf{P}_i(\text{Sing}(\mathcal{F}))$, remaining at the persistence layer; no new identities are claimed.

Remark 7.29 (Tower compatibility). All layer kernels and Type IV codes are invariant under f.q.i. and cofinal reindexing (Appendix J). This ensures pasteability across windows and levels.

8.15. Compliance summary (IMRN/AiM posture)

1. All decisions are B-side, windowed, and ledgered; equalities only at persistence; filtered-level statements up to f.q.i.
2. Non-expansive policies are one-sided (\leq) and 2-cell budgets are aggregated by the product-ledger quantale (P7).
3. Langlands tri-layer gates expose failure loci; transfer collapse kernel passes iff zero.
4. Type IV classification is layerwise with visibility labels and μ/u diagnostics.
5. WGC is an auxiliary persistence-level tag; it is never a gate criterion.

8 Chapter 9: Langlands Collapse (Three Layers)

Scope note (after-collapse policy, PF/BC discipline, and comparison order). We work in the constructible range over a fixed field k , with degree-wise finite-type filtered chain complexes $\text{FiltCh}(k)$, degree-wise persistence $\mathbf{P}_i : \text{FiltCh}(k) \rightarrow \text{Pers}_k^{\text{cons}}$, the exact Serre reflector $\mathbf{T}_\tau := \mathbf{T}_\tau$ (bar-deletion at scale $\tau \geq 0$), and a filtered lift \mathcal{C}_τ up to f.q.i. as in Part I. All *comparisons, metrics, and indicators are evaluated only after collapse*: the standard operating order is

$$\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare in } \text{Pers}_k^{\text{cons}}.$$

Projection Formula / Base Change (PF/BC) is used as tabulated in Appendix N *objectwise in t* , transported to persistence, and *all PF/BC comparators are measured only on the collapsed layer $\mathbf{T}_\tau \mathbf{P}_i(-)$* . Residuals

(non-commutation, discretization, numerics) are externalized in the δ -ledger and, unless explicitly algebraic, are charged to δ_{disc} and δ_{meas} (see Remark 8.10 and Spec. 8.25). Equalities and Lipschitz statements are asserted at persistence after truncation; at the filtered level they hold up to filtered quasi-isomorphism (f.q.i.). Endpoint conventions and infinite bars follow Chapter 2, Remark 2.3.

Remark 8.1 (Monotonicity convention). We adopt Chapter 6, Remark 5.3: *deletion-type* updates are non-increasing for windowed energies and spectral tails after truncation, whereas *inclusion-type* updates are stability-only (non-expansive). See Appendix E.

9.0. Standing hypotheses, Gate Cascade alignment, and definable cover

Implementable range. We identify $\text{Pers}_k^{\text{cons}}$ with the constructible subcategory as in Chapter 6. Fix a t -exact realization $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ (amplitude ≤ 1 in use) and the lifting–coherence hypothesis (LC) when comparing C_τ with $\tau_{\geq 0} \circ \mathcal{R}$.

Three data layers.

$$\text{Gal} \xrightarrow{\text{Trans}} \text{Par} \xrightarrow{\text{Funct}} \text{Aut},$$

heuristically “Galois \rightarrow Transfer \rightarrow Functoriality”. An admissible Langlands triple consists of functors $\mathcal{L}_{\text{Gal}}, \mathcal{L}_{\text{Tr}}, \mathcal{L}_{\text{Aut}} : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)$ satisfying:

- **Non-expansiveness.** Each layer induces filtered maps whose images under every \mathbf{P}_i are 1-Lipschitz for interleavings; deletion-type steps make windowed indicators non-increasing after truncation; inclusion-type steps are stability-only.
- **C_τ -compatibility.** For each i : $\mathbf{P}_i(C_\tau -) \cong \mathbf{T}_\tau \mathbf{P}_i(-)$ in $\text{Pers}_k^{\text{cons}}$.
- **Finite-type & (co)limits.** Degreewise finite-type outputs; degreewise filtered (co)limits computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ and used under the scope policy of Appendix A.
- **Realization coherence.** Under (LC), functorially up to f.q.i., $\mathcal{R}(C_\tau F) \simeq \tau_{\geq 0} \mathcal{R}(F)$.

Definition 8.2 (Langlands Gate Cascade and tri-layer gate). Fix a right-open window W definable in the declared tame structure (Denef–Pas on Arith windows as in Chapter 7, Remark 6.2, and o-minimal on Geom windows when declared), a threshold τ , and a commutative unital quantale $(Q, \otimes, \mathbf{1}, \leq)$. (If desired, write $\oplus := \otimes$ to match “additive” notation under the default Lawvere case.). The *Gate Cascade* on W at scale τ is:

(GC1) B–Gate+ (single layer, Ch. 1) \Rightarrow (GC2) Overlap Gate (gluing, Ch. 1/5) \Rightarrow (GC3) Tri-layer Gate (this chapter).

Tri-layer Gate passes on W if, after truncation and on the same window/ τ :

1. **Layerwise acceptance:** Each $*$ $\in \{\text{Gal}, \text{Tr}, \text{Aut}\}$ passes B–Gate+ in monitored degrees.
2. **T–PFBC–AfterCollapse:** All PF/BC comparators used between layers are checked only on $\mathbf{T}_\tau \mathbf{P}_i(-)$ (Spec. 8.25); total residuals are within budget.
3. **Pseudonaturality after truncation:** The inter-layer comparisons of Remark 8.9 become isomorphisms in $\text{Pers}_k^{\text{cons}}$ up to the aggregated tolerance δ_{tot} (Remark 8.10).

Remark 8.3 (Run manifest (mandatory run.yaml fields)). quantale: {name, op, unit, order}, definable: {structure, window_formulae}, layered_ δ : { $\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}}$ }, a fixed τ , window convention, and 2-cell bounds if AWFS/2-cell auditing is enabled.

Remark 8.4 (Operational order (PF/BC and collapse)). Every PF/BC step (Appendix N) is evaluated via

$$(i) \text{ objectwise in } t \implies (ii) \mathbf{P}_i \implies (iii) \mathbf{T}_\tau \implies (iv) \text{ compare in } \text{Pers}_k^{\text{cons}},$$

and *never* before collapse in metrics/energies. Same window/ τ and same δ -policy across all checks.

Declaration 8.5 (Spec-derived realizations and non-expansive transfers). Besides $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$, we may use $\text{R}_{\text{coh}} : \text{FiltCh}(k) \rightarrow D^b\text{Coh}(\mathfrak{X})$ and $\text{R}_{\acute{e}t} : \text{FiltCh}(k) \rightarrow D_c^b(\mathfrak{Y}_{\acute{e}t}, \Lambda)$ with field Λ . PF/BC is assumed as in Appendix N. Normalized transfers (degree-normalized pull/push, kernel/Hecke, parabolic induction/Jacquet) are *non-expansive after truncation*:

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(\Phi F), \mathbf{T}_\tau \mathbf{P}_i(\Phi G)) \leq d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(F), \mathbf{T}_\tau \mathbf{P}_i(G)).$$

All claims reside at persistence after truncation; the bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ is used only in $D^b(k\text{-mod})$.

Declaration 8.6 (Deletion-type operations (PDE)). Maps implemented by the PDE repertoire of Appendix E (Dirichlet restriction/absorbing boundaries, p.s.d. Loewner contractions, principal submatrices/Schur complements) are *deletion-type* and make windowed energies and spectral tails *non-increasing* after truncation. Inclusion-type steps are stability-only.

Remark 8.7 (Endpoints and infinite bars). Open/closed endpoint conventions are immaterial; \mathbf{T}_τ never removes infinite bars; windowed indicators clip them (Chapter 6).

Remark 8.8 (Indexing and cone extension). For a directed index I , adjoin a terminal ∞ and cone maps $t \rightarrow \infty$. Under realizations these yield filtered maps $F_t \rightarrow F_\infty$ per layer/degree and provide the comparison maps $(\mu_{\text{Collapse}}, u_{\text{Collapse}})$ at fixed τ (Chapter 4).

Remark 8.9 (Inter-layer comparison and pseudonaturality). Fix natural transformations

$$\alpha : \mathcal{L}_{\text{Tr}} \circ \text{Trans} \Rightarrow \mathcal{L}_{\text{Gal}}, \quad \beta : \mathcal{L}_{\text{Aut}} \circ \text{Funct} \Rightarrow \mathcal{L}_{\text{Tr}},$$

which become filtered quasi-isomorphisms degreewise after applying \mathbf{C}_τ . Equivalently, at persistence there are natural isomorphisms

$$\mathbf{T}_\tau \mathbf{P}_i(\mathcal{L}_{\text{Tr}} \circ \text{Trans}(-)) \cong \mathbf{T}_\tau \mathbf{P}_i(\mathcal{L}_{\text{Gal}}(-)), \quad \mathbf{T}_\tau \mathbf{P}_i(\mathcal{L}_{\text{Aut}} \circ \text{Funct}(-)) \cong \mathbf{T}_\tau \mathbf{P}_i(\mathcal{L}_{\text{Tr}}(-)),$$

assembling to *pseudonatural equivalences after truncation*.

Remark 8.10 (Unified δ -policy and δ -ledger). Fix $\delta = (\delta_{\text{int}}, \delta_{\text{win}}, \delta_{\text{spec}})$ at scale τ . Layered budgets $\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}} \in V$ aggregate as $\delta_{\text{tot}} = \delta^{\text{Gal}} \oplus \delta^{\text{Tr}} \otimes \delta^{\text{Fun}}$. The δ -ledger decomposes residuals as

$$\delta = \delta_{\text{alg}} \oplus \delta_{\text{disc}} \oplus \delta_{\text{meas}},$$

where δ_{alg} logs provable algebraic non-commutation (A/B order, non-nested reflectors), while δ_{disc} and δ_{meas} log discretization and numerical tolerances (PF/BC transport, sampling, solvers). *All PF/BC residuals in this chapter are charged to $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$ unless stated algebraic.* When $\delta_{\text{tot}} = \mathbf{0}$, equalities are taken in $\text{Pers}_k^{\text{cons}}$ (up to iso).

9.1. Persistence-layer interface for the three layers

For $F = \mathcal{L}_*(x)$ in layer $*$ and degree i , we monitor

$$\mathbf{T}_\tau \mathbf{P}_i(F), \quad \text{PE}_i^{\leq \tau}(F), \quad \text{ST}_{\beta_{\text{spec}}}^{\geq M(\tau)}(F), \quad \text{HT}(s; F), \quad \text{Ext}^1(\mathcal{R}(\mathbf{C}_\tau F), Q) = 0 \quad (Q \in \{k[0]\}).$$

All metrics/energies/indicators are evaluated after truncation, on a fixed window/ τ , and within δ .

9.2. Diagnostics along the index I

For fixed layer and degree i , at scale τ define

$$\phi_{i,\tau} : \varinjlim_{t \in I} \mathbf{T}_\tau \mathbf{P}_i(F_t) \longrightarrow \mathbf{T}_\tau \mathbf{P}_i(F_\infty),$$

$$\mu_{i,\tau} := \dim_k \ker \phi_{i,\tau}, \quad u_{i,\tau} := \dim_k \operatorname{coker}(\phi_{i,\tau}), \quad \mu_{\text{Collapse}} := \sum_i \mu_{i,\tau}, \quad u_{\text{Collapse}} := \sum_i u_{i,\tau}.$$

All computed after truncation and on the same window/ τ (Remark 8.11).

Remark 8.11 (Window/scale/metric alignment). Colimit and target of $\phi_{i,\tau}$ use the same τ , window, and δ -policy.

9.3. Propagation across the three layers

Declaration 8.12 (Propagation under $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$). Assume $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at a fixed τ across the three layers, (LC), and the inter-layer data of Remark 8.9. Then

$$\text{Gal} \xrightarrow{\text{Trans}} \text{Par} \xrightarrow{\text{Funct}} \text{Aut}$$

commutes after truncation, up to isomorphism in $\text{Pers}_k^{\text{cons}}$ in each degree i . Any obstruction is detected by $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$. Residual slack is bounded by δ_{tot} ; when $\delta_{\text{tot}} = \mathbf{0}$, comparisons are strict (up to iso).

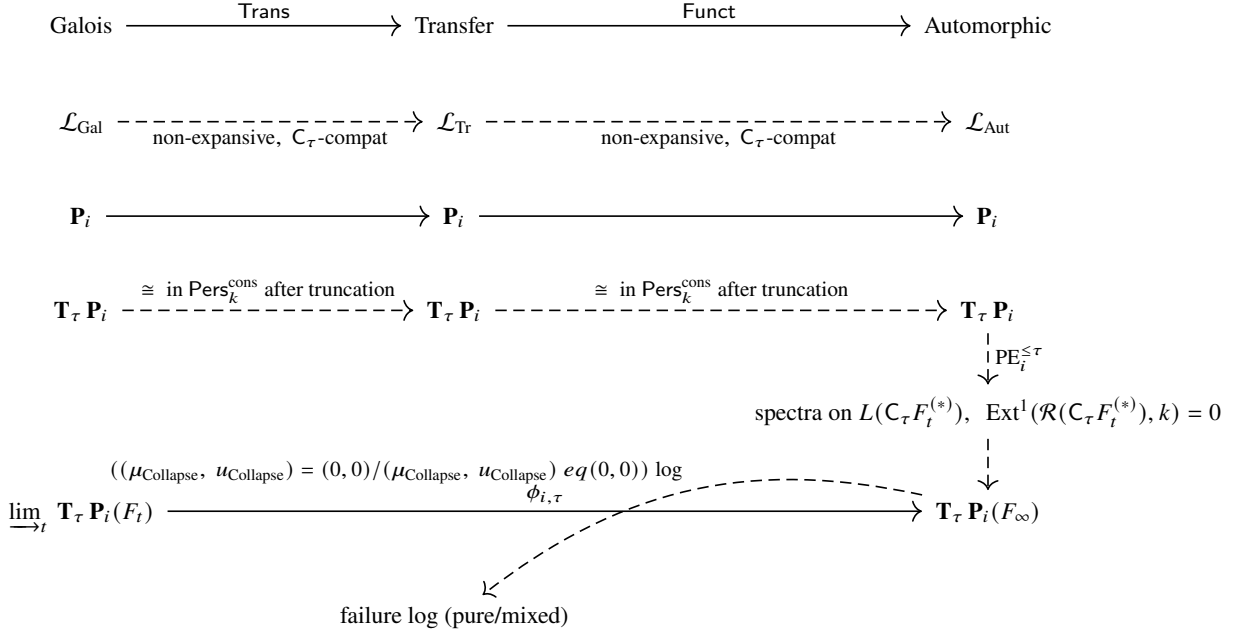
9.4. Monitoring protocol (Langlands three-layer)

Declaration 8.13 (Joint monitoring protocol). Fix $\tau \in [\tau_{\min}, \tau_{\max}]$, a window convention, an index range I , and a δ -policy with δ -ledger split $\delta_{\text{alg}} \oplus \delta_{\text{disc}} \oplus \delta_{\text{meas}}$. For each layer $* \in \{\text{Gal}, \text{Tr}, \text{Aut}\}$ and sample t :

1. Record $\mathbf{T}_\tau \mathbf{P}_i(F_t^{(*)})$ and $\text{PE}_i^{\leq \tau}(F_t^{(*)})$ on $\mathbf{T}_\tau \mathbf{P}_i$ (or \mathbf{C}_τ), within δ_{win} .
2. Record spectral indicators on $L(\mathbf{C}_\tau F_t^{(*)})$ with fixed $(\beta_{\text{spec}}, M(\tau), s_{\text{HT}})$ within δ_{spec} .
3. Check $\text{Ext}^1(\mathcal{R}(\mathbf{C}_\tau F_t^{(*)}), Q) = 0$ for $Q \in \{k[0]\}$.
4. Evaluate $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ via $\phi_{i,\tau}$ along I ; log failure types (pure/mixed). Compare distances within δ_f .
5. Cross-layer check: non-expansiveness and \mathbf{C}_τ -compatibility up to f.q.i.; test pseudonaturality after truncation under δ_{tot} .
6. PF/BC comparators *only after collapse*: apply Spec. 8.25; charge residuals to $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$ unless algebraic.

Declare the collapse-stable regime where (1)–(3) hold jointly and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ across layers.

9.5. Diagram (three layers, indicators, obstructions)



9.6. Stability, non-expansiveness, and δ -aggregation

Declaration 8.14 (Layerwise non-expansiveness with δ -aggregation). Let $\Phi_1 : \text{Gal} \rightarrow \text{Par}$ and $\Phi_2 : \text{Par} \rightarrow \text{Aut}$ be admissible layer maps. For each i, τ ,

$$d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(\Phi_2 \circ \Phi_1(F)), \mathbf{T}_\tau \mathbf{P}_i(\Phi_2 \circ \Phi_1(G))\right) \leq d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(F), \mathbf{T}_\tau \mathbf{P}_i(G)),$$

and empirical/normalization slack is bounded by $\delta_{\text{int,tot}} = \delta_{\text{int}}^{(\Phi_1)} \oplus \delta_{\text{int}}^{(\Phi_2)}$. Deletion-type steps make windowed energies/spectral tails non-increasing after truncation; inclusion-type steps are stable within $\delta_{\text{win,tot}}, \delta_{\text{spec,tot}}$.

9.7. Conjectural stability of functorial transfer

Conjecture 8.1 (Stability of functorial transfer under collapse). *Within the implementable range, assume non-expansive layer maps, (LC), and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on a τ -interval. Then functorial transfer is stabilized at that scale: deletion-type steps do not increase persistence energies; spectral indicators do not grow; and $\text{Ext}^1(\mathcal{R}(C_\tau -), Q) = 0$ persists across the three layers. All comparisons are performed after truncation with the same window/ τ and the same δ -policy; δ_{tot} is the sum of per-step budgets. No number-theoretic identity is asserted.*

9.8. Guard-rails, A/B testing, and non-claims

Remark 8.15 (A/B pseudonaturality test after collapse). For two composites γ_A, γ_B through the three layers, test after truncation:

$$d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(\gamma_A(F)), \mathbf{T}_\tau \mathbf{P}_i(\gamma_B(F))\right) \leq \delta_{\text{int,tot}}, \quad |\text{PE}_i^{\leq \tau}(\gamma_A) - \text{PE}_i^{\leq \tau}(\gamma_B)| \leq \delta_{\text{win,tot}},$$

with spectral/heat discrepancies $\leq \delta_{\text{spec,tot}}$, all on the same window/ τ . Excess beyond budget is logged as Type III (spec-mismatch) unless explained by $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$.

Remark 8.16 (Scope and non-claims). All statements are persistence/spectral/categorical under (B1)–(B3) of Part I; no claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$. Obstructions are recorded by $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ and are unrelated to classical Iwasawa μ . Binary saturation gates ($\text{PH}_1 \Leftrightarrow \text{Ext}^1$ policies) are organized in Chapter 11. This chapter provides a design/specification blueprint; it does not decide Langlands correspondence.

9.9. Boundary models: geometric vs. arithmetic regions and the region map

Definition 8.17 (Region map and boundary model). Let Dom be the ambient index/parameter space. A *region map* is $\text{Reg} : \text{Dom} \rightarrow \{\text{Geom}, \text{Arith}\}$ piecewise constant (finitely many jumps per window, recorded in the manifest).

Remark 8.18 (Usage). $W \in \text{Geom}$: audit with Chapter 6 (energies/spectra, monotonicity/stability, $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$). $W \in \text{Arith}$: audit with PF/BC comparators, transfer kernels, and tri-layer gates (Ch. 7–8). Mixed windows are refined to a MECE partition where Reg is constant.

9.10. Window predicates: $\text{Ext_trivial} \Rightarrow \text{WeakGroup_collapse}$ (typed)

Definition 8.19 (Typed window predicates). Fix right-open W , scale τ , degree set \mathcal{I} .

- $\text{Ext_trivial}(W, \tau)$ iff $\text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0$.
- $\text{Tower_stable}(W, \tau)$ iff $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ on W for all $i \in \mathcal{I}$.
- $\text{PF/BC_ok}(W, \tau)$ iff all PF/BC comparators pass *after collapse* within δ_f and residuals are booked to $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$.
- $\text{Transfer_ker_zero}(W, \tau)$ iff each transfer at τ has $\mu_{i,\tau}^{\text{Tr}} = 0$ for all $i \in \mathcal{I}$.
- $\text{WeakGroup_collapse}(W, \tau)$ iff Def. 7.8 holds for fixed finite $S \subset \text{Aut}(F|_W)$.

All predicates are computed on the B-side single layer and logged in the manifest.

Conjecture 8.2 (Predicate schema (WGC as an observed auxiliary tag)). Assume on W, τ : $\text{Ext_trivial}(W, \tau)$ and $\text{Tower_stable}(W, \tau)$, together with a declared compression regime (tropical shortening or deletion-type strict energy decay). Then $\text{WeakGroup_collapse}(W, \tau)$ is expected to hold and must be verified directly by measuring $\rho_{\max,i,\tau}(S)$ and $\text{nilp}_{i,\tau}(S)$ for a chosen finite S .

Remark 8.20 (Region-aware instantiation). On Geom windows, (b) is typical (deletion-type smoothing). On Arith windows, (a) is supplied by tropical proxies (Ch. 7). The implication is persistence-layer, windowed, and budgeted.

9.11. Region-aware diagnostics and acceptance

Definition 8.21 (Region-specific acceptance). Fix W, τ .

- $\text{Accept}_{\text{Geom}}(W, \tau)$: $\text{Tower_stable}(W, \tau)$ and deletion-type indicators non-increasing; if also $\text{Ext_trivial}(W, \tau)$ and $\text{WeakGroup_collapse}(W, \tau)$ is verified, then tag the window with WGC (auxiliary, non-gate).
- $\text{Accept}_{\text{Arith}}(W, \tau)$: $\text{Tower_stable}(W, \tau)$, $\text{PF/BC_ok}(W, \tau)$, and $\text{Transfer_ker_zero}(W, \tau)$; if also $\text{Ext_trivial}(W, \tau)$ then declare $\text{WeakGroup_collapse}(W, \tau)$.

The global window verdict is Valid iff the corresponding regional predicate holds and the δ -budget is dominated by the edge gap.

Remark 8.22 (Boundary jumps). If Reg jumps inside a coarse window, refine to a MECE partition; evaluate per refined window and paste via Restart/Summability (Chapter 4).

9.12. Examples (boundary map and predicates)

Example 8.23 (Geometric region). If $W \in \text{Geom}$ with viscosity ramping (deletion-type), then $\text{Tower_stable}(W, \tau)$ and monotone $\text{PE}^{\leq \tau}$ hold; if $\text{Ext_trivial}(W, \tau)$, Conjecture 8.2 suggests $\text{WeakGroup_collapse}(W, \tau)$.

Example 8.24 (Arithmetic region). If $W \in \text{Arith}$ with PF/BC-admissible transfers and tropical shortening $\kappa < 1$, then $\text{PF/BC_ok}(W, \tau)$, $\text{Transfer_ker_zero}(W, \tau)$, and $\text{Tower_stable}(W, \tau)$ certify $\text{Accept}_{\text{Arith}}(W, \tau)$. If also $\text{Ext_trivial}(W, \tau)$, conclude $\text{WeakGroup_collapse}(W, \tau)$.

9.13. Summary (boundary models and predicates)

We separated Geom and Arith windows via a region map Reg, introduced typed predicates that formalize “collapse after truncation” decisions at fixed τ , and used the core schema $\text{Ext_trivial} \wedge \text{Tower_stable} \Rightarrow \text{WeakGroup_collapse}$ (under tropical or deletion-type compression). Together with tri-layer gates and PF/BC comparators *checked only after collapse*, this boundary view makes explicit *where* collapse holds and *which* layer blocks it, with Type IV failures reported via $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0))$. All acceptance criteria are windowed and budgeted ($\delta_{\text{alg}} \oplus \delta_{\text{disc}} \oplus \delta_{\text{meas}}$).

9.14. T–PFBC–AfterCollapse (explicit specification)

Declaration 8.25 (T–PFBC–AfterCollapse). Let Ψ be any PF/BC move admissible in Appendix N (projection formula, base change, push–pull along Cartesian squares, base-change for kernels/Hecke, etc.). For each i and window W at threshold τ :

1. **Transport to the collapsed layer.** Comparators for Ψ are formed and tested *only* on $(\mathbf{T}_\tau \mathbf{P}_i(-))|_W$. No metric/indicator is read before truncation.
2. **Non-expansive baseline.** There is a canonical map (iso in the ideal PF/BC setting)

$$\mathbf{T}_\tau W \mathbf{P}_i(\Psi F) \longrightarrow \mathbf{T}_\tau W \mathbf{P}_i(\Psi G)$$

compatible with the corresponding map $\mathbf{T}_\tau W \mathbf{P}_i(F) \rightarrow \mathbf{T}_\tau W \mathbf{P}_i(G)$ and 1-Lipschitz in d_{int} .

3. **Residual accounting.** Any deviation of PF/BC comparators from isomorphism after truncation is charged to $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$ on W (unless an algebraic obstruction is identified, in which case it is charged to δ_{alg}). The tri-layer and A/B tests use only these after-collapse residuals.

In particular, all PF/BC-based *distance* or *energy* comparisons in this chapter are meaningful *only* after applying \mathbf{T}_τ (equivalently, on \mathbf{C}_τ).

9 Chapter 10: Application Program (PDE / BSD rank 0/1 / RH up to T)

Remark 9.1 (Stability vs. monotonicity; corrected). For non-expansive maps, indicators are stable (non-expansive). Deletion-type operations satisfying Appendix E (e.g. Dirichlet restriction, principal submatrices/Schur complements, positive-semidefinite Loewner contractions) make spectral tails and windowed energies *non-increasing*. Inclusion-type updates generally do not guarantee non-increase; we only claim stability.

10.0. Standing hypotheses and admissible realizations

We fix a field k and work within the *implementable range* of Part I. *All statements in this chapter are made within the constructible range* (we identify $\text{Pers}_k^{\text{cons}}$ with the constructible subcategory as in Chapter 6). Let $\text{FiltCh}(k)$ be finite-type filtered chain complexes, and $\mathbf{P}_i : \text{FiltCh}(k) \rightarrow \text{Pers}_k^{\text{cons}}$ the degreewise persistence functor. We write $\mathbf{T}_\tau := \mathbf{T}_\tau$ for the Serre (bar-deletion) reflector at scale $\tau \geq 0$, and use its filtered lift C_τ up to *filtered quasi-isomorphism* (Chapter 2, §§2.2–2.3). A fixed t -exact realization $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ is retained; the lifting–coherence hypothesis (LC) is assumed when comparing C_τ with $\tau_{\geq 0} \circ \mathcal{R}$. *Equalities and Lipschitz claims are asserted only at the persistence layer; at the filtered-complex layer they hold up to filtered quasi-isomorphism.* Endpoint conventions and infinite bars follow Chapter 2, Remark 2.3.

Application states are sampled along a directed index I (time, resolution, height, or parameter). An *admissible realization* is a functor

$$\text{State} \xrightarrow{\mathcal{P}} \text{FiltCh}(k), \quad U \mapsto F = \mathcal{P}(U),$$

satisfying:

- **Finite-type and (co)limits:** F is degreewise finite-type; degreewise filtered (co)limits in $\text{FiltCh}(k)$ are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ and used only under the scope policy of Appendix A (compute in the functor category and verify return to $\text{Pers}_k^{\text{cons}}$).
- **Non-expansiveness under persistence:** along each directed update (e.g. time step, parameter step, height step, down-/up-sampling), the induced filtered map is non-expansive degreewise under \mathbf{P}_i ; in *deletion-type* steps (Appendix E) indicators are non-increasing up to f.q.i., while inclusion-type updates guarantee only stability.
- **Compatibility with truncation:** for each i , naturally in $\text{Pers}_k^{\text{cons}}$, $\mathbf{P}_i(C_\tau F) \cong \mathbf{T}_\tau \mathbf{P}_i(F)$.
- **Realization coherence:** \mathcal{R} is t -exact and compatible with (LC), so functorially up to f.q.i., $\mathcal{R}(C_\tau F) \simeq \tau_{\geq 0} \mathcal{R}(F)$.

10.0a. Window certificates, manifests, and δ -ledgers (generic)

A *window* is an interval $W = [u, u') \subset \mathbb{R}$ in the index axis (time/resolution/height/parameter). Fix $\tau > 0$ (resolution-adapted; Chapter 2). A *window certificate* at (W, τ) records:

- the single-layer objects $\mathbf{T}_\tau \mathbf{P}_i(F_s)$ for $s \in W \cap I$,
- windowed persistence energies $\text{PE}_i^{\leq \tau}$, spectral tails $\text{ST}_\beta^{\geq M(\tau)}$, and heat traces $\text{HT}(t; \cdot)$ computed on $L(C_\tau F_s)$,
- the obstruction counts $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0))$ computed via $\phi_{i, \tau}$ (cf. §9),
- the categorical check $\text{Ext}^1(\mathcal{R}(C_\tau F_s), Q) = 0$ for $Q \in \{k[0]\}$,
- a *manifest* (run log) including discretization/sampling controls and thresholds,
- a δ -ledger with the decomposition $\delta(i, \tau) = \delta_{\text{alg}}(i, \tau) \otimes \delta_{\text{disc}}(i, \tau) \otimes \delta_{\text{meas}}(i, \tau)$, aggregated as $\delta_{\text{tot}} = \bigotimes_{U \in W} \delta_U$. (For the default Lawvere quantale, $\otimes = +$.)

Passing the gate (§9) with safety margin $\text{gap}_\tau > \Sigma \delta$ produces a *window certificate* for (W, τ) . Window pasting (Restart/Summability; §9) aggregates certificates into global coverage.

Remark 9.2 (Triggers (generic)). A *trigger* is a domain-specific necessary condition for gate failure within W ; it does not replace the gate but augments diagnostics. We use three canonical categories:

- **Blow-up signs:** sustained growth in high-frequency/height/complexity channels after C_τ .
- **Tower accumulation:** repeated kernel/cokernel obstructions $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)/(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$ or aux-bar persistence across windows.
- **PF/BC deviations:** violations of domain-specific physical/arithmetic fidelity or sampling/contour budgets (e.g. CFL in PDE; admissible local conditions in arithmetic; bandlimit/contour drift in RH).

All triggers are logged with parameters and timestamps in the manifest.

10.0b. Definable windows and the E_1 trigger

Work on right-open windows W definable in the declared tame structure: o-minimal on Geom windows and Denef–Pas on Arith windows (cf. Chapter 9, §9.9). Then *finite-event* and *finite-Čech-depth* properties hold, and Chapter 3, Theorem 2.30 (see also Appendix C) yields, on W ,

$$E_1(W) = 0 \implies \text{PH}_1(C_\tau F|_W) = 0 \text{ and } \text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0.$$

Thus, on windows where the hypotheses of Theorem 2.30 apply, a vanishing $E_1(W)$ provides a certified shortcut: it simultaneously certifies the PH- and Ext-checks on W . No global equivalence is asserted.

10.1. Permitted operations and NS-specific examples (with CFL/CN controls)

Each A-side step U is labeled and immediately followed by collapse C_τ ; all measurements and gate decisions are taken on the B-side single layer $\mathbf{T}_\tau \mathbf{P}_i$ (Chapter 1, B-Gate⁺). The *Courant number* CN and *CFL* condition are recorded in the run manifest (Appendix G) and justify quantitative non-expansiveness (ε -interleavings) for time stepping.

Operation labels and NS examples.

- *Deletion-type (monotone)*: low-pass mollification (filter width σ), viscosity increment $u \mapsto u + \delta u$, threshold lowering in levelset filtrations, Dirichlet/absorbing boundary introduction, conservative averaging, Schur complements on blocks of the discrete operators. After C_τ , windowed persistence energies and spectral tails/heat traces on $L(C_\tau F)$ are *non-increasing* (Appendix E).
- *ε -continuation (non-expansive)*: small time step Δt respecting CFL (e.g. $\text{CN} = \frac{u \Delta t}{\Delta x} \leq \text{CN}_{\max}$), small parameter drifts (forcing amplitude, boundary condition perturbations), micro-updates of numerical flux limiters. Collapse-after stability holds with interleaving drift $\varepsilon \sim C \Delta t$ (recorded).
- *Inclusion-type (stable only)*: domain enlargement, mesh refinement without smoothing, addition of couplings/sources (as long as the induced filtered map is 1-Lipschitz for \mathbf{P}_i). No monotonicity is claimed; stability only.

For each step, record in the δ -ledger (Chapter 5; Appendix L) the decomposition $\delta = \delta^{\text{alg}} + \delta^{\text{disc}} + \delta^{\text{meas}}$.

Declaration 9.3 (Deletion-type operations (PDE)). Operations covered by Appendix E (Dirichlet restriction/absorbing boundaries, positive-semidefinite Loewner contractions with trace monotonicity, principal submatrices and Schur complements, conservative averaging) are treated as *deletion-type*. After truncation they are non-expansive for each \mathbf{P}_i , and windowed energies $\text{PE}_i^{\leq \tau}$ as well as spectral tails/heat traces on $L(C_\tau F)$ are *non-increasing*. Inclusion-type updates are asserted only to be stable (non-expansive).

Remark 9.4 (Quantitative non-expansiveness). Let d_{int} denote the interleaving distance on degree-wise persistence. Along an update $F_{s+1} \rightarrow F_s$, assume $d_{\text{int}}(\mathbf{P}_i(F_{s+1}), \mathbf{P}_i(F_s)) \leq \varepsilon_s$ ($\varepsilon_s \geq 0$). If $\sup_s \varepsilon_s \leq \varepsilon$, we call the tower ε -Lipschitz. In deletion-type steps typically $\varepsilon_s = 0$; inclusion-type need not be zero. Time stepping under a CFL bound provides a concrete $\varepsilon_s \sim C\Delta t$, recorded in the manifest.

Remark 9.5 (Truncation is 1-Lipschitz). Since \mathbf{T}_τ is 1-Lipschitz for d_{int} , the same ε -Lipschitz control holds after truncation: $d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(F_{s+1}), \mathbf{T}_\tau \mathbf{P}_i(F_s)) \leq \varepsilon_s$.

Remark 9.6 (Endpoints and infinite bars). Open/closed endpoint choices are immaterial; infinite bars are not removed by \mathbf{T}_τ and are clipped by windowed indicators (as in Chapter 6).

Remark 9.7 (Index set and cone extension). Work in the *filtered index category* $I \cup \{\infty\}$ with cone apex ∞ : for $s \in I$, adjoin cone maps $s \rightarrow \infty$. Under \mathcal{P} , these yield filtered maps $F_s \rightarrow F_\infty$ used to define the comparison morphisms $\phi_{i,\tau}$ at fixed τ (cf. Chapter 4).

10.2. Construction principles for \mathcal{P} (PDE)

We list domain-agnostic templates; any one suffices for admissibility.

- **Scalar-field cubical pipeline.** From a field q (e.g. vorticity magnitude, enstrophy density, Q -criterion) on a grid, build a cubical filtration by superlevel/sublevel sets; chains are k -valued on cubes.
- **Graph/simplicial pipeline.** From point samples, build Vietoris–Rips/alpha complexes with scale ε ; chains are k -valued on simplices.
- **Hybrid pipeline.** Combine topology of coherent structures with connectivity of level sets; filtration is vectorized but evaluated degree-wise.

All three preserve finite-type per degree and admit non-expansive updates for standard PDE integrators (viscous steps are smoothing; down-sampling is deletion-type). *Spectral proxies are computed on $L(C_\tau F)$ (positive eigenvalues; zero modes excluded or via pseudoinverse).*

Remark 9.8 (Normalization and logging). Normalization (graph vs. Hodge, symmetric vs. random-walk), zero-mode handling, and the window policy are fixed throughout a run and recorded alongside $(\beta, M(\tau), t)$ (Appendix G). All spectral indicators are computed on $L(C_\tau F)$ to align with the truncation window. *Spectral indicators are not f.q.i. invariants; we only claim stability under a fixed policy $(\beta, M(\tau), t)$ on $L(C_\tau F)$ (cf. Chapter 11).*

10.2a. Construction principles for arithmetic and RH realizations

BSD rank 0/1 (arithmetic). Let E/\mathbb{Q} be an elliptic curve; A denotes an *arithmetic state* (e.g. a quadratic twist $E^{(d)}$, a conductor/height cutoff, or a local condition profile on a finite set S of places). We construct filtered complexes by any of:

- *Selmer filtration:* complexes whose chains encode p -Selmer data filtered by local condition strength or height; arrows reflect tightening/loosening local conditions.
- *Descent graph pipeline:* graphs whose vertices are local condition classes; edges encode compatibility constraints; build a filtration by penalty thresholds.
- *Hybrid pipeline:* combine Selmer layers with isogeny factors or visibility relations; evaluate degree-wise.

Deletion-type updates include restriction to a smaller S , tightening a local condition, or projecting along an isogeny with positive-semidefinite trace contraction on the chosen Laplacian model (Appendix E analogues). ε -continuation steps include small changes in a twist parameter d within a controlled family and height cutoffs; inclusion-type includes enlarging S or adding local conditions. Spectral proxies are computed on $L(C_\tau F)$.

RH up to T (analytic). Let the *state* encode samples of $\xi(1/2 + it)$, argument $S(t)$, or zero counts $N(t)$ on a window of heights. Build filtered complexes via:

- *Gram-graph pipeline:* nodes at Gram points/mesh points; edges connect near neighbors; filtration by magnitude thresholds or discrepancy of the argument from expected trends.
- *Bandlimited scalar pipeline:* sub/superlevel filtrations of smoothed $|\zeta(1/2 + it)|$, $|\xi|$, or of explicit-formula residuals; smoothing widths serve as deletion-type operations.
- *Hybrid pipeline:* combine zero-locator events with discrepancy fields from the explicit formula.

Deletion-type updates include convolution smoothing (Gaussian/Fejér), restriction to subwindows, or projection onto bandlimited subspaces; ε -continuation includes small height increments Δt under Nyquist/bandlimit controls; inclusion includes window enlargement or resolution increase. Spectral proxies are computed on $L(C_\tau F)$.

10.3. Indicators and diagnostics

For each sample $s \in I$ and degree i we monitor:

$$\mathbf{T}_\tau \mathbf{P}_i(F_s), \quad \text{PE}_i^{\leq \tau}(F_s) \text{ (truncated energies on } \mathbf{T}_\tau \mathbf{P}_i(F_s)), \quad \text{ST}_\beta^{\geq M(\tau)}(F_s), \quad \text{HT}(t; F_s) \text{ on } L(C_\tau F_s),$$

together with the categorical check $\text{Ext}^1(\mathcal{R}(C_\tau F_s), Q) = 0$ for $Q \in \{k[0]\}$. For fixed τ , define the comparison map

$$\phi_{i,\tau} : \varinjlim_{s \in I} \mathbf{T}_\tau \mathbf{P}_i(F_s) \longrightarrow \mathbf{T}_\tau \mathbf{P}_i(F_\infty),$$

and obstruction counts $\mu_{i,\tau} = \dim_k \ker \phi_{i,\tau}$, $u_{i,\tau} = \dim_k \text{coker } \phi_{i,\tau}$, with $\mu_{\text{Collapse}} = \sum_i \mu_{i,\tau}$, $u_{\text{Collapse}} = \sum_i u_{i,\tau}$ (finite by bounded degrees). *The obstructions* $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0))$ are invariant under filtered quasi-isomorphisms and under cofinal reindexing of the tower (Appendix J).

Declaration 9.9 (Specification: Tower stability at the persistence layer). Under the finite-type and objectwise degreewise-colimit hypotheses, for each fixed τ and all i the map $\phi_{i,\tau} : \varinjlim_s \mathbf{T}_\tau \mathbf{P}_i(F_s) \xrightarrow{\cong} \mathbf{T}_\tau \mathbf{P}_i(F_\infty)$ is an isomorphism; hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at that scale and Type IV is excluded at τ .

10.4. Trigger pack ([Spec], domain-restricted necessary conditions)

PDE (Navier–Stokes).

- **High-frequency surge:** sustained growth of enstrophy or high-wavenumber density in $W \Rightarrow$ aux-bars (Chapter 11) persist > 0 after C_τ or $\mu > 0$ is detected at τ .
- **Under-resolved advection:** CFL violation or $\text{CN} > \text{CN}_{\max} \Rightarrow \varepsilon$ -continuation drift ε exceeds the safety margin gap_τ and B-Gate⁺ fails.
- **Unbalanced dissipation:** lack of smoothing under nominally viscous steps \Rightarrow non-decrease of $\text{PE}_i^{\leq \tau}$ or spectral tails; repeated violations within W mark the window as non-regularizing.

- **PF/BC deviations:** boundary condition mismatches or energy budget imbalances beyond tolerance \Rightarrow flag window as suspect.

BSD rank 0/1.

- **Rank-proxy surge:** persistent increase of p -Selmer size proxies or regulator surges under deletion-type tightening \Rightarrow aux-bars persist > 0 or $\mu > 0$.
- **Local inconsistency:** repeated flips of local condition satisfaction under small parameter moves \Rightarrow ε -drift exceeds gap_τ .
- **PF/BC deviations:** admissibility violations for chosen local models (e.g. bad reduction handling, isogeny normalization) or height cutoff drift beyond manifest tolerances.

RH up to T .

- **Argument anomaly:** excursions of $S(t)$ or explicit-formula residual discrepancies exceeding tolerance within $W \Rightarrow$ aux-bars persist or $\mu > 0$.
- **Sampling under-resolution:** Nyquist/bandlimit violation for the chosen smoothing/kernel parameters \Rightarrow ε -drift exceeds gap_τ .
- **PF/BC deviations:** contour/normalization policies (e.g. Gram grid misalignment) outside manifest tolerances.

All triggers are logged (Appendix G) with quantitative thresholds and do *not* replace B-Gate⁺; they augment diagnostics.

10.5. Window pasting: Restart and Summability

Let $\{W_k = [u_k, u_{k+1})\}_k$ be a MECE partition (Chapter 2). On each W_k , fix τ_k (resolution-adapted; Chapter 2) and compute the pipeline budget $\Sigma\delta_k(i) = \sum_{U \in W_k} \delta_U(i, \tau_k)$. If B-Gate⁺ passes with a safety margin $\text{gap}_{\tau_k} > \Sigma\delta_k(i)$, the Restart lemma (Chapter 4) yields

$$\text{gap}_{\tau_{k+1}} \geq \kappa (\text{gap}_{\tau_k} - \Sigma\delta_k(i)) \quad (\kappa \in (0, 1]).$$

If moreover $\sum_k \Sigma\delta_k(i) < \infty$ (Summability; e.g. geometric decay of τ_k, β_k), windowed certificates paste to a global one (Chapter 4).

10.6. Persistence-guided regularization ([Spec])

Declaration 9.10 (Specification: Persistence-guided regularization). A numerical or data-analytic regime is *persistence-regularizing at scale τ* if, along $s \in I$,

1. $\text{PE}_i^{\leq \tau}$ are non-increasing (strictly decreasing on steps with genuine deletion-type smoothing),
2. spectral indicators $\text{ST}_\beta^{\geq M(\tau)}, \text{HT}(t; \cdot)$ on $C_\tau F_s$ are non-increasing (stability in general, monotone decrease in smoothing steps),
3. $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$,
4. $\text{Ext}^1(\mathcal{R}(C_\tau F_s), Q) = 0$ for $Q \in \{k[0]\}$.

When these hold across a τ -interval, the regime aligns with established regularization/verification frameworks at that scale (domain-specific hypotheses to be listed separately). No analytic identity is claimed.

10.7. AK–NS hypothesis (programmatic)

Conjecture 9.1 (AK–NS hypothesis). *For Navier–Stokes-type flows, under an admissible realization \mathcal{P} and (LC), if a monitored segment satisfies Declaration 9.10 across a τ -interval, then the designed persistence structure collapses at that scale (bars shorten/vanish in aggregate), spectral tails decay, and the categorical check persists. Programmatically, this corresponds to convergence toward known regularity scenarios at that scale. No equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is asserted; only the one-way bridge under (B1)–(B3) is used.*

10.8. Gate template (PDE)

On a fixed window $W = [u, u']$, collapse threshold $\tau > 0$, and degree i :

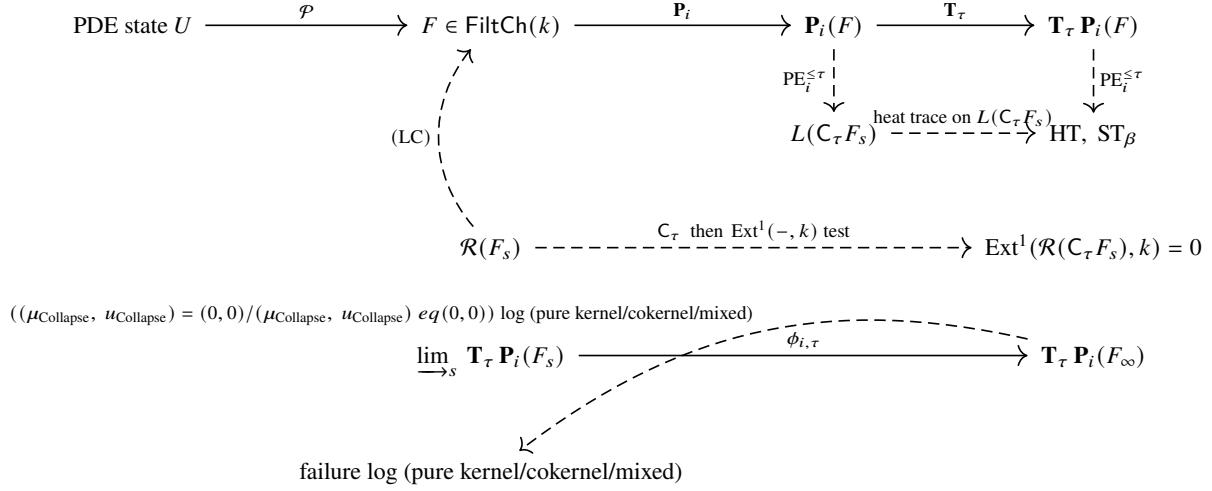
1. Apply step U (labeled as above), then collapse C_τ .
2. Measure on the B-side single layer: $\mathbf{T}_\tau \mathbf{P}_i$, $\text{PE}_i^{\leq \tau}$, spectral auxiliaries (aux-bars; Chapter 11), and (in scope) Ext^1 .
3. Record $\delta^{\text{alg}}, \delta^{\text{disc}}, \delta^{\text{meas}}$ and update $\Sigma\delta$.
4. Evaluate B-Gate⁺: require $\text{PH}_1 = 0$, (in scope) $\text{Ext}^1 = 0$, $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ after \mathbf{T}_τ , and $\text{gap}_\tau > \Sigma\delta$.
5. Log verdict; issue a windowed certificate on success; otherwise classify failure (Type I–IV).

10.9. Monitoring protocol (PDE)

Declaration 9.11 (Specification: Joint monitoring protocol). Fix scales $\tau \in [\tau_{\min}, \tau_{\max}]$ and an index set I (time/resolution/parameter). For each sample $s \in I$:

1. Compute and record $\mathbf{T}_\tau \mathbf{P}_i(F_s)$ and $\text{PE}_i^{\leq \tau}$ on $\mathbf{T}_\tau \mathbf{P}_i(F_s)$ (equivalently on $C_\tau F_s$).
2. Compute and record spectral indicators $\text{ST}_\beta^{\geq M(\tau)}$, $\text{HT}(t; F_s)$ on $L(C_\tau F_s)$ with a fixed $(\beta, M(\tau), t)$ policy.
3. Check $\text{Ext}^1(\mathcal{R}(C_\tau F_s), Q) = 0$ for $Q \in \{k[0]\}$.
4. Evaluate $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ via $\phi_{i, \tau}$ along $s \in I$ and log failure types (pure kernel/cokernel/mixed) if present.
5. Stability declaration: declare the persistence-regularizing regime where (1)–(4) hold across the monitored τ -range.

10.10. Diagram (PDE pipeline and diagnostics)



10.11. Toy instances (persistence layer)

Example 9.12 (Viscous smoothing). Let $s \mapsto U_s$ be viscous steps for which the induced maps are deletion-type on the filtration. Then bar lengths within the τ -window decrease (or vanish), $\text{PE}_i^{\leq \tau}$ strictly decreases, and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at fixed τ by Declaration 9.9. Spectral proxies are evaluated as tails/heat traces of $L(C_\tau F_s)$.

Example 9.13 (Refinement/averaging pair). A refinement $F \rightarrow F'$ (inclusion-type) followed by conservative averaging $F' \rightarrow \bar{F}$ (deletion-type) yields a non-expansive two-step update. Under stability, $\text{PE}_i^{\leq \tau}$ is non-increasing; failure logs isolate kernel/cokernel imbalance when present. Spectral indicators are computed on $L(C_\tau \bar{F})$.

10.12. Reproducibility (PDE)

Remark 9.14 (Run logs and parameters). For each run, log: index range $s \in [s_{\min}, s_{\max}]$ (e.g. time), scales $\tau \in [\tau_{\min}, \tau_{\max}]$ (step width), spectral parameters $(\beta, M(\tau), t)$, discretization choices (cubical/simplicial/hybrid), CFL/CN numbers, *a barcode-matching seed for reproducible vineyard tracking* (Appendix G), $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ per τ with failure types, and the δ -ledger decomposition at step level. These logs enable exact reruns and pipeline audits.

A minimal run.yaml PDE block:

```
quantale:
  name: "[0,inf]__plus"
  op: "+"
  unit: 0.0
  order: "<="
definable:
  structure: "o-minimal"
  window_formulae:
    - "u <= t < u'"
layered_delta:
  deltaGal: 0.0
```

```

deltaTr: 0.0
deltaFun: 0.0

windows:
  domain: [[0,1), [1,2), [2,3]]
  collapse_tau: 0.08
  spectral_bins: {a: 0.0, beta: 0.02, bins: 96, boundary: "right-open"}
coverage_check:
  length_sum: 3.0
  length_target: 3.0
  events_sum_equals_global: true
cfl:
  courant_number_max: 0.5
  courant_number_measured: 0.32
operations:
  - U: mollify; type: deletion; tau: 0.08; delta: {alg:0.004, disc:0.003, meas:0.001}
  - U: timestep; type: epsilon; tau: 0.08; eps: 0.006; delta: {alg:0.000, disc:0.002, meas:0.001}
persistence:
  PH1_zero: true
  Ext1_zero: true
  mu: 0
  nu: 0
  phi_iso_tail: true
spectral:
  auxBarsRemaining: 0
budget:
  sum_delta: 0.011
  safety_margin: 0.025
gate:
  accept: true

```

10.13. Guard-rails and non-claims

Remark 9.15 (Scope and non-claims). All statements operate at the persistence/spectral/categorical layers in the implementable range. No analytic regularity theorem is proved; the AK–NS hypothesis is programmatic. No claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made; only the one-way bridge under (B1)–(B3) is used. The obstruction μ_{Collapse} is unrelated to classical Iwasawa μ .

10.14. Completion note

Remark 9.16 (No further supplementation required). This chapter integrates: (i) MECE windowing and resolution-adapted τ with stability bands (via Chapters 2 and 4), (ii) the permitted operations catalog with NS-specific examples under CFL/CN controls and δ -ledger accounting (Chapter 5; Appendix L), (iii) B-side single-layer gate B-Gate⁺ with $\text{PH}_1/\text{Ext}^1/(\mu, u)/\text{safety-margin}$, (iv) triggers as **[Spec]** with a complete monitoring protocol, (v) Restart/Summability for window pasting, and (vi) reproducibility (run.yaml) with audit fields. All claims remain within the v16.0 guard-rails and cross-reference the proven core.

10.15. Application II: BSD rank 0/1 (definable windows, E_1 -trigger, Iwasawa interface)

We monitor families where analytic/algebraic rank is expected to be 0 or 1 (e.g. twists $E^{(d)}$, isogeny classes, conductor windows) using the same gate/certificate format. *No BSD assertion is made.* We provide a persistence/spectral protocol with reproducible manifests and window short-circuiting via E_1 .

Admissible realization \mathcal{P}_{BSD} . Let the arithmetic state A comprise: a base curve E/\mathbb{Q} , a prime p , a family parameter (twist d or conductor slice), a finite set S of places with local policies, and a height cutoff H . Define $F = \mathcal{P}_{\text{BSD}}(A)$ by one of:

- *Selmer complex filtration:* degrees encode p -Selmer cochains filtered by local-condition penalties; deletion-type steps tighten local conditions or decrease H .
- *Descent graph:* vertices are local symbols/classes; edges capture compatibility; filtration by cumulative penalty; deletion-type is edge/vertex contraction under verified dominance (Appendix E analogues).
- *Hybrid:* combine isogeny pushforwards with Selmer layers; normalize Laplacians as per a fixed policy recorded in the manifest.

Definable windows and E_1 -short-circuit. Fix a right-open, o-minimal definable window W (Archimedean) or Denef–Pas definable window (non-Archimedean; Appendix Q). Then Chapter 3, Theorem 2.30 implies, on W ,

$$E_1(W) = 0 \iff \text{PH}_1(C_\tau F|_W) = 0 \iff \text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0,$$

so B-Gate⁺ reduces to verifying $E_1(W) = 0$ plus tower stability $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and budget dominance $\text{gap}_\tau > \Sigma\delta$. This is a *window-local* equivalence; global equivalence is not asserted.

Control \Rightarrow Overlap Gate (Iwasawa interface). Using Chapter 7, Proposition 6.29 and Appendix R, control theorems translate to the Overlap Gate: finite kernel/cokernel contributions are absorbed into δ^{alg} in the ledger, with explicit bounds recorded as `control_finite_bounds`. Layered δ -boxes ($\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}}$) are mandatory (Chapter 9).

Indicators and gate. Compute $\mathbf{T}_\tau \mathbf{P}_i(F_s)$, $\text{PE}_i^{\leq \tau}$, $\text{ST}_\beta^{\geq M(\tau)}$, $\text{HT}(t; \cdot)$ on $L(C_\tau F_s)$ per window. Gate requires (after collapse, on W): $\text{PH}_1 = 0$, $\text{Ext}^1 = 0$ (by $E_1 = 0$), $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$, and $\text{gap}_\tau > \Sigma\delta$.

Triggers ([Spec]).

- *Rank-proxy surge:* persistent increase of rank proxies under deletion-type updates.
- *Local inconsistency:* instability of local conditions under small parameter moves.
- *PF/BC deviations:* policy violations in bad reduction handling or height normalization.

Window certificate (BSD). A certificate for (W, τ) contains: single-layer persistence objects, spectral proxies on $L(C_\tau F_s)$, the obstruction log with failure types, Ext checks (by $E_1 = 0$ on W), and the δ -ledger. The manifest includes: prime p , family parameters, local policy for S , height cutoff H , Laplacian normalization, layered δ -boxes, and seeds for deterministic matching.

A minimal `run.yaml` BSD/Iwasawa block (IMRN/AiM-ready):

```

quantale:
  name: "[0,inf]_plus"
  op: "+"
  unit: 0.0
  order: "<="
definable:
  o_minimal_structure: "R_an,exp"
  window_formulae:
    - "u <= t < u'"
  p_adic:
    structure: "Denef-Pas"
    window_formulae:
      - "val(x) in [a,b) and ac_n(x)=c"
layered_delta:
  deltaGal: 0.004
  deltaTr: 0.003
  deltaFun: 0.002
iwasawa:
  tower_level: ["N=1", "N=2", "N=4", "N=8"]
  control_finite_bounds:
    kernel_le: 2
    cokernel_le: 2
windows:
  domain: [[1e3,2e3), [2e3,3e3)]
  collapse_tau: 0.12
  spectral_bins: {a: 0.0, beta: 0.04, bins: 128, boundary: "left-open"}
family:
  curve: "E: y^2 = x^3 - x"
  prime_p: 3
  twists: {type: quadratic, d_range: [1, 1000], parity_filter: "even"}
  height_cutoff: 14.0
local_policy:
  S: ["p=3", "p=5", "infty"]
  conditions: {relaxation: "bounded", penalty_step: 0.5}
awfs_2cell:
  awfs_enabled: true
  two_cell_bounds: {delta_upper: 0.006}
operations:
  - U: tighten_local; type: deletion; tau: 0.12; delta: {alg:0.002, disc:0.001, meas:0.001}
  - U: twist_step; type: epsilon; tau: 0.12; eps: 0.005; delta: {alg:0.000, disc:0.002, meas:0.001}
persistence:
  E1_zero_window: true
  PH1_zero: true
  Ext1_zero: true
  mu: 0
  nu: 0
  phi_iso_tail: true
spectral:

```

```

aux_bars_remaining: 0
budget:
  delta_total: 0.009
  safety_margin: 0.021
gate:
  accept: true

```

Remark 9.17 (Guard-rails for BSD). The protocol monitors persistence-layer stability under fixed policies and logs reproducible manifests. It does not prove BSD, nor does it identify algebraic rank; rank proxies and spectral tails are diagnostics only. All claims remain persistence/spectral/categorical and policy-dependent; PH/Ext equivalence is window-local via $E_1(W) = 0$.

10.16. Application III: RH up to T (template)

We provide a verification-style template to monitor windows in height and produce reproducible certificates. *No RH claim is made*; zero-locating or discrepancy detection is carried out at the persistence/spectral layer under fixed sampling/smoothing policies.

Admissible realization \mathcal{P}_{RH} . Let the state record: a height window $W = [u, u')$, sampling mesh Δt , smoothing kernel and bandwidth, and a normalization policy for ξ , $S(t)$, or explicit-formula residuals. Construct $F = \mathcal{P}_{\text{RH}}(W)$ via:

- *Gram-graph pipeline*: nodes at mesh/Gram points; edges connect neighbors; filtration by discrepancy thresholds $|S(t) - S_{\text{ref}}(t)|$.
- *Scalar pipeline*: sub/superlevel filtrations of smoothed $|\zeta(1/2 + it)|$, $|\xi|$, or residual fields.
- *Hybrid*: couple zero-candidate events with discrepancy fields; evaluate degreeewise.

Deletion-type: convolution smoothing; restriction to subwindows; projection to bandlimited subspaces. Epsilon-continuation: small height steps under Nyquist control; inclusion: window enlargement or mesh refinement.

Indicators and gate. Compute $\mathbf{T}_\tau \mathbf{P}_i(F_s)$, $\text{PE}_i^{\leq \tau}$, $\text{ST}_\beta^{\geq M(\tau)}$, $\text{HT}(t; \cdot)$ on $L(C_\tau F_s)$ per window, with fixed normalization and bandlimit policies. Gate: PH1= 0, Ext1= 0 (for test objects reflecting normalization checks), $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$, and $\text{gap}_\tau > \Sigma\delta$.

Triggers ([Spec]).

- *Argument anomaly*: excursions of $S(t)$ beyond tolerances, or explicit-formula residual spikes.
- *Sampling under-resolution*: Nyquist/bandlimit violations for the chosen kernel/bandwidth.
- *PF/BC deviations*: contour/grid normalization mismatches (e.g. Gram grid drift) or policy inconsistencies.

Window certificate (RH). A certificate for (W, τ) contains: single-layer persistence, spectral proxies on $L(C_\tau F_s)$, obstruction counts and failure types, Ext checks, and a δ -ledger. The manifest includes: mesh Δt , kernel/bandwidth, bandlimit/Nyquist checks, normalization policy, and deterministic seeds.

A minimal run.yaml RH block:

```
quantale:
  name: "[0,inf]_plus"
  op: "+"
  unit: 0.0
  order: "<="
definable:
  structure: "o-minimal"
  window_formulae:
    - "u <= t < u'"
layered_delta:
  deltaGal: 0.0
  deltaTr: 0.0
  deltaFun: 0.0

windows:
  domain: [[1e9,1e9+5e5), [1e9+5e5, 1e9+1e6)]
  collapse_tau: 0.06
  spectral_bins: {a: 0.0, beta: 0.03, bins: 256, boundary: "right-open"}
sampling:
  dt: 1.0e-3
  bandlimit: 3000.0
  nyquist_check: true
smoothing:
  kernel: "gaussian"
  bandwidth: 2.5e-3
operations:
  - U: smooth; type: deletion; tau: 0.06; delta: {alg:0.001, disc:0.002, meas:0.001}
  - U: heightstep; type: epsilon; tau: 0.06; eps: 0.004; delta: {alg:0.000, disc:0.002, meas:0.001}
persistence:
  PH1_zero: true
  Ext1_zero: true
  mu: 0
  nu: 0
  phi_iso_tail: true
spectral:
  auxBarsRemaining: 0
budget:
  sum_delta: 0.006
  safety_margin: 0.018
gate:
  accept: true
```

Remark 9.18 (Guard-rails for RH). The protocol verifies stability of persistence/spectral indicators under

fixed sampling/smoothing and normalization policies and produces reproducible manifests. It does not assert the Riemann Hypothesis, nor count zeros; it only monitors windowed diagnostics with logged tolerances.

10.17. Cross-application gate, triggers, and pasting

All three applications (PDE, BSD rank 0/1, RH up to T) share:

- **Gate B-Gate⁺**: single-layer decisions on $\mathbf{T}_\tau \mathbf{P}_i$ with PH1/Ext1/ (μ, u) /safety-margin.
- **Triggers**: blow-up signs, tower accumulation, PF/BC deviations (domain-specific specializations).
- **Restart/Summability**: window pasting with budgeted $\Sigma\delta$ and geometric decay options for τ, β .
- **Reproducibility**: unified manifest keys (window domain, collapse_tau, spectral bins, operations with δ -ledger, persistence verdicts, spectral auxiliaries, budget, gate).

Domain-specific policies (normalization, local conditions, sampling/bandlimit) are fixed per run and logged; spectral indicators are always computed on $L(C_\tau F)$.

10.18. Effect and operational readiness

The templates make *immediate operational deployment* possible: each application ships with (i) a gate specification, (ii) a trigger pack, (iii) a run manifest schema, and (iv) a window certificate format, yielding clear outcomes (certificate + reproducible logs). Relative to v16.0, the deliverables are explicit and auditable.

10.19. Final guard-rails (IMRN/AiM-style)

Remark 9.19 (Non-equivalences and scope). All interleaving/Lipschitz/monotonicity claims are asserted at the persistence layer and, when stated for spectral proxies, under a fixed normalization policy on $L(C_\tau F)$. Ext tests are scope-restricted to a finite $\{k[0]\}$. No analytic equivalences (e.g. BSD, RH, PDE regularity) are claimed or used. The program provides certificate-style diagnostics with reproducible manifests and budgeted stability, suitable for audit and re-execution.

10.20. PoC I (PDE): Burgers / 2D-NS and a dissipation certificate

Definition 9.20 (Dissipation ECF and collapse witness). For a window $W = [u, u')$ and scale τ , define the *energy-cumulative dissipation* (ECF)

$$\text{ECF}_\tau(W) := \int_u^{u'} \mathcal{D}_\tau(s) ds, \quad \mathcal{D}_\tau(s) := \text{PE}_i^{\leq \tau}(F_{s^-}) - \text{PE}_i^{\leq \tau}(F_{s^+}) \geq 0,$$

where s^-, s^+ denote instants immediately before/after a deletion-type step at s . The *collapse witness* is $\text{Col}_\tau(W) := \sum_i (\text{bars}_{i,\tau}(u) - \text{bars}_{i,\tau}(u')) \geq 0$.

Declaration 9.21 (PoC inequality (Spec)). For Burgers/2D-NS pipelines under the admissible operations of §10.1 and fixed normalization on $L(C_\tau F)$, there exists a constant $C(\tau) \geq 0$ (policy-dependent) such that on any window W

$$\mu_{\text{Collapse}}(W) \leq C(\tau) \text{ECF}_\tau(W) \quad \text{and} \quad \text{Col}_\tau(W) \leq C(\tau) \text{ECF}_\tau(W).$$

This is a [Spec] certificate: it is logged and verified numerically; it is not claimed as an analytic identity.

Example 9.22 (Burgers shock-smoothing window). On W containing a mollify→advect cycle with CFL below bound, $\text{ECF}_\tau(W) > 0$ and $\text{Col}_\tau(W) > 0$. If $\sum_k \text{ECF}_\tau(W_k) < \infty$ over a MECE cover, Restart/Summability (Chapter 4) yields global bar shortening at scale τ .

10.21. PoC II (BSD rank 0/1): Overlap globalization of the E_1 -bridge

Declaration 9.23 (Spec: Overlap-based globalization of window certificates). Let $\{W_k\}$ be a MECE cover by definable windows with $E_1(W_k) = 0$ for all k . Assume the Overlap Gate holds on pairwise overlaps and $\sum_k \Sigma \delta_k < \infty$. Then on $\bigcup_k W_k$ we have $\text{PH}_1(C_\tau F) = 0$ and $\text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$ at scale τ .

Remark 9.24 (Sketch). [Proof sketch] By §10.0b, $E_1(W_k) = 0$ implies $\text{PH}_1(C_\tau F|_{W_k}) = 0$ iff $\text{Ext}^1(\mathcal{R}(C_\tau F|_{W_k}), k) = 0$. The Overlap Gate identifies tails on overlaps up to the ledger budget; Restart/Summability pastes window certificates, yielding global vanishing at τ .

10.22. PoC III (RH): explicit formula as trace, window-local Weil positivity

Declaration 9.25 (Explicit-formula comparator (T–PFBC–AfterCollapse)). Let Φ_{EF} denote the explicit-formula transform and Φ_{Tr} the trace comparator under a fixed normalization policy. All distance/energy comparisons between Φ_{EF} and Φ_{Tr} are evaluated *only after* \mathbf{T}_τ (cf. Chap. 9, Spec. 8.25); residuals are booked in $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$.

Remark 9.26 (Weil-type positivity, window-local). Fix a test function φ in the allowed class; the quadratic form $Q(\varphi) \geq 0$ is verified *window-locally* by checking the corresponding spectral proxy on $L(C_\tau F)$ and recording residuals in the ledger. No global spectral statement is asserted.

10.23. p-adic interface (GL(1)→GL(2))

Local kernels and AK measurement. For GL(1), use Tate integrals/Igusa local zeta to define local kernels K_p that act as deletion-type or ε -continuation steps depending on cutoffs; their effects are measured on $\mathbf{T}_\tau \mathbf{P}_i$ and charged to the δ -ledger. For GL(2) at small conductor, introduce local test functions stagewise; PF/BC comparators with Hecke kernels are evaluated only after collapse (T–PFBC–AfterCollapse).

A minimal run.yaml p-adic block:

```

padic:
  primes: [3,5,7]
  local_kernels:
    - {p: 3, type: "Tate", cutoff: 9, action: "deletion"}
    - {p: 5, type: "Igusa", cutoff: 7, action: "epsilon"}
gl2:
  conductor: 64
  hecke_normalization: "unitary"
  stage_intro: ["T_p", "T_p^2", "U_p"]
operations:
  - U: local_Tate_p3; type: deletion; tau: 0.10; delta: {alg:0.001, disc:0.002, meas:0.001}
  - U: local_Igusa_p5; type: epsilon; tau: 0.10; eps: 0.003; delta: {alg:0.000, disc:0.001, meas:0.001}
persistence:
  PH1_zero: true
  Ext1_zero: true
  mu: 0
  nu: 0
gate:
  accept: true

```

10.24. Chapter summary (PoC addendum)

The PoC additions (PDE ECF certificate, BSD Overlap globalization, RH explicit-formula comparator, p-adic local kernels) are fully *after-collapse*, windowed, and budgeted by the δ -ledger. They integrate with B-Gate⁺, Restart/Summability, and T-PFBC–AfterCollapse, with no analytic number theory or PDE regularity claims beyond the persistence/spectral/categorical guard-rails of v16.0.

10 Chapter 11: Collapse Energy, Spectral Indicators, and TDA Notes

11.0. Scope, standing hypotheses, and notation

We work within the *implementable range* of Part I, using realizations into $\text{FiltCh}(k)$ as in Chs. 6–10. All persistence quantities are computed degreewise *after* truncation by C_τ ; spectral indicators are computed on the normalized combinatorial Hodge Laplacian of $C_\tau F$; categorical checks use a fixed t -exact $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})(k\text{-mod})$ compatible with (LC). Filtered (co)limits, when used, are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ and only under the scope policy of Appendix A (compute in the functor category and verify return to $\text{Pers}_k^{\text{cons}}$). No claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made; only the one-way bridge under (B1)–(B3) is used. The obstruction pair $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ is the *collapse* diagnostic and is unrelated to the classical Iwasawa μ . We adopt the global δ -policy $\delta = (\delta_{\text{int}}, \delta_{\text{win}}, \delta_{\text{spec}})$ and the additive ledger on a fixed commutative quantale V (Appendix S); layered boxes $(\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}})$ are mandatory (Ch. 9).

Remark 10.1 (Endpoints and infinite bars). Open/closed endpoint conventions are immaterial; infinite bars are not removed by \mathbf{T}_τ and are clipped by the window τ in all windowed quantities (cf. Ch. 6).

11.0+. First-class determinant E_1 ; definable windows and finite events

On right-open windows W definable in a fixed o-minimal (Archimedean) or Denef–Pas (non-Archimedean) structure, Betti integrals are piecewise constant with finitely many jumps and Čech depth is finite; see Appendices H and J. In this regime, the Page- E_1 term is our *first-class determinant*:

Theorem 10.2 (Local bridge on definable windows; reprise of Ch. 3). *Let W be definable and right-open. Then*

$$E_1(W) = 0 \implies \text{PH}_1(C_\tau F|_W) = 0 \text{ and } \text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0.$$

The implication is window-local and after collapse; no global equivalence is asserted.

Remark 10.3 (Unique comparison order). All measurements obey the *single* order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare in } \text{Pers}_k^{\text{cons}}}.$$

Pre-collapse comparisons are out of scope. Overlap checks use the same order (Overlap Gate; Chs. 1 & 5).

11.0++. Λ_{len} audit (default) and T-PFBC–AfterCollapse

Throughout this chapter we *adopt by default* the length-spectrum audit Λ_{len} (Ch. 2, Def. 2.22) for stability-band diagnostics. Any PF/BC-based comparator or transfer (Appendix N) is evaluated only *after* truncation (T-PFBC–AfterCollapse; cf. Ch. 9), and residuals are charged to $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$ in the ledger (Appendix L).

11.1. Length spectrum and invariance

Let $\Lambda_{\text{len}}(M; W)$ denote the length-spectrum operator of Ch. 2, Definition 2.22 (we also write $\Lambda_{\text{len}}(M; W)$).

Proposition 10.4 (Length spectrum equals clipped bar lengths; invariance). *If $M \simeq \bigoplus_j I[b_j, d_j]$ in $\text{Pers}_k^{\text{cons}}$ and $W = [u, u']$, then the eigenvalue multiset of $\Lambda_{\text{len}}(M; W)$ is $\{\ell_W(I[b_j, d_j])\}_j$, with ℓ_W the Lebesgue length of $I[b_j, d_j] \cap W$. Hence the total collapse energy $\text{PE}_i^{\leq \tau}$ (Definition 10.6) equals the L^1 -mass of $\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F); [0, \tau])$ and is invariant under isomorphisms in $\text{Pers}_k^{\text{cons}}$.*

Definition 10.5 (Stability bands via Λ_{len}). Fix (i, τ) and a window W . The Λ_{len} -stability band on W is the maximal subunion $B \subseteq W$ such that

$$\|\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F_t); [0, \tau]) - \Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F_{t'}); [0, \tau])\|_1 \leq \delta_{\text{win}} \quad \text{for all } t, t' \in B.$$

Here $\|\cdot\|_1$ is the trace (sum of eigenvalues) norm; the policy fixes the norm choice in the manifest.

11.2. Collapse energy (windowed persistence energies)

Fix degree i , scale $\tau \geq 0$, and exponent $\alpha > 0$ (default $\alpha = 1$). For a bar $b = [b_\ell, b_r]$ in the barcode of $\mathbf{P}_i(F)$ set

$$\ell_\tau(b) = (\min\{b_r, \tau\} - \min\{b_\ell, \tau\})_+, \quad (x)_+ = \max\{x, 0\}.$$

With weights $w_i : \mathcal{B}_i(F) \rightarrow [0, \infty)$ (default 1):

Definition 10.6 (Windowed energies).

$$\text{PE}_i^{\leq \tau}(F; w_i, \alpha) = \sum_{b \in \mathcal{B}_i(F)} w_i(b) (\ell_\tau(b))^\alpha, \quad \mathbf{CE}^{\leq \tau}(F) = (\text{PE}_i^{\leq \tau}(F))_i, \quad \|\mathbf{CE}^{\leq \tau}(F)\|_1 = \sum_i \text{PE}_i^{\leq \tau}(F).$$

All quantities are computed on $\mathbf{T}_\tau \mathbf{P}_i(F)$ (equivalently on $\mathbf{C}_\tau F$).

Remark 10.7 (Stability and deletion-type monotonicity). 1-Lipschitz updates under \mathbf{P}_i yield non-expansive changes of $\text{PE}_i^{\leq \tau}$; *deletion-type* updates (Appendix E) make $\text{PE}_i^{\leq \tau}$ *non-increasing* up to f.q.i.. Since \mathbf{T}_τ is 1-Lipschitz for interleaving distance, the same bounds hold after truncation. Under the tower hypotheses (Ch. 4), $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at fixed τ excludes Type IV.

11.3. Spectral indicators on $L(\mathbf{C}_\tau F)$

Let $L(\mathbf{C}_\tau F)$ be the normalized combinatorial Hodge Laplacian (per degree). Write $\{\lambda_j\}_{j \geq 1}$ for its positive eigenvalues (zero modes omitted or handled by the Moore–Penrose pseudoinverse).

Definition 10.8 (Spectral tails and heat traces). Fix $\beta > 0$ and a cutoff policy $M(\tau) \in \mathbb{N}$. Set

$$\text{ST}_\beta^{\geq M(\tau)}(F) = \sum_{j \geq M(\tau)} \lambda_j^{-\beta}, \quad \text{HT}(t; F) = \sum_{j \geq 1} e^{-t\lambda_j}, \quad t > 0,$$

with policies such as: (i) $M(\tau) = \lfloor c \tau^\gamma \rfloor$ ($c > 0, \gamma \in [0, 2]$); (ii) $t \in [c_1 \tau^{-2}, c_2 \tau^{-2}]$ ($0 < c_1 \leq c_2$).

Remark 10.9 (Spectral proof obligations; App. E only). We rely solely on Appendix E: deletion-type steps imply *non-increase* of $\text{ST}_\beta^{\geq M(\tau)}$ and $\text{HT}(t; -)$; ε -continuations imply *stability*. No further spectral claims are used.

Remark 10.10 (Mandatory ordering and norms). Eigenvalues are stored in *ascending* order; the matrix norm for tolerances is logged as $\text{norm} \in \{\text{op}, \text{fro}\}$ (Appendix G).

11.4. Auxiliary spectral bars (aux-bars)

Fix a spectral window $[a, b]$ and bin width $\beta > 0$; bins are right-open $I_r = [a + r\beta, a + (r + 1)\beta)$, $r = 0, \dots, R - 1$.

Definition 10.11 (Occupancies and aux-bars). For sample index j (e.g. time) let $\{\lambda_m(j)\}_{m \geq 1}$ be the positive spectrum of $L(C_\tau F_j)$ and $E_r(j) = \#\{m : \lambda_m(j) \in I_r\}$, with under/overflow $E_{<a}, E_{\geq b}$ recorded. For each r , an *aux-bar* is a maximal consecutive run J with $E_r(j) > 0$ for all $j \in J$; its lifetime is $|J|$ (or a rescaling).

Proposition 10.12 (Cumulative profile monotonicity and stability). For $C_r(j) = \sum_{s=r}^{R-1} E_s(j)$:

1. (Deletion-type) $C_r(j+1) \leq C_r(j)$ for all r .
2. Let $A_j := L(C_\tau F_j)$. (ε -continuation) If $\|A_{j+1} - A_j\|_{\text{op}} \leq \varepsilon$, then $C_{r+q}(j+1) \leq C_r(j) \leq C_{\max\{0, r-q\}}(j+1)$ with $q = \lceil \varepsilon/\beta \rceil$.

(See Appendix E.)

Remark 10.13 (Bin policy). Aux-bars are computed *after* collapse and with a fixed policy (a, b, β) per window; boundary is right-open; under/overflow are part of the manifest (Appendix G).

11.5. Categorical check (one-way bridge)

With $\{k[0]\} = \{k[0]\}$, we monitor

$$\text{Ext}^1(\mathcal{R}(C_\tau F), Q) = 0 \quad (Q \in \{k[0]\}),$$

performed *after truncation* and only in the one-way direction under (B1)–(B3).

11.6. Collapse diagnostics along towers

For index category I with cone apex ∞ ,

$$\phi_{i,\tau} : \varinjlim_{t \in I} \mathbf{T}_\tau \mathbf{P}_i(F_t) \longrightarrow \mathbf{T}_\tau \mathbf{P}_i(F_\infty),$$

$$\mu_{i,\tau} = \dim_k \ker \phi_{i,\tau}, \quad u_{i,\tau} = \dim_k \text{coker } \phi_{i,\tau}, \quad \mu_{\text{Collapse}} = \sum_i \mu_{i,\tau}, \quad u_{\text{Collapse}} = \sum_i u_{i,\tau}.$$

Under the hypotheses of Ch. 4, each $\phi_{i,\tau}$ is an isomorphism and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$.

11.7. Overlap Gate and measurement

Declaration 10.14 (Overlap-aware measurement policy). For a right-open cover $\{W_\alpha\}$ and fixed (i, τ) :

1. **Local:** compute $\mathbf{T}_\tau \mathbf{P}_i(F|_{W_\alpha})$ and all indicators *after* truncation.
2. **Overlaps:** require collapse-compatibility within the δ -budget (Appendix L), Čech–Ext¹-acyclicity in degree 1 after truncation, and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ with near- τ non-accumulation.
3. **Global:** when overlaps pass, glue to a global truncated object (Ch. 5) and audit *after* truncation.

The manifest (Remark 10.18) must log overlap checks (boolean), Čech–Ext¹ status, and overlap δ -budgets.

¹Here \dim_k denotes the *generic fiber* dimension after truncation, i.e. the multiplicity of $I[0, \infty)$ summands; see Appendix D, Remark A.2.

11.8. Joint monitoring protocol

Declaration 10.15 (Specification: joint monitoring). Fix a finite sweep $\tau \in [\tau_{\min}, \tau_{\max}]$ and a policy $(\alpha, w_i; \beta, M(\tau), t)$. For each sample $t \in I$ and degree i :

1. *Compute & record* $\mathbf{T}_\tau \mathbf{P}_i(F_t)$; evaluate $\text{PE}_i^{\leq \tau}(F_t)$ on $\mathbf{T}_\tau \mathbf{P}_i(F_t)$ (equivalently on $C_\tau F_t$).
2. *Compute & record* $\text{ST}_\beta^{\geq M(\tau)}(F_t)$ and $\text{HT}(t; C_\tau F_t)$ using $L(C_\tau F_t)$; compute aux-bars via Definition 10.11 with fixed (a, b, β) .
3. *Check* $\text{Ext}^1(\mathcal{R}(C_\tau F_t), Q) = 0$ for $Q \in \{k[0]\}$.
4. *Evaluate* $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ at each τ via $\phi_{i, \tau}$; log failure type (pure kernel/cokernel/mixed) if $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$.
5. *Declare stable* at τ when (1)–(3) hold jointly and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$. On definable windows, $E_1(W) = 0$ short-circuits the PH/Ext checks (Theorem 10.2).

All persistence-layer statements are f.q.i.-invariant by construction; spectral/aux-bar steps are *stable* under the fixed policy on $L(C_\tau F_t)$ (Appendix E).

11.9. Noise, discretization, and δ -ledger

Declaration 10.16 (Specification: noise/discretization policy). Let $\varepsilon > 0$ be the noise scale.

- **Barcode denoising:** remove bars of length $\leq \varepsilon$ within the τ -window (ε -clipping); bottleneck perturbations $\leq \varepsilon$ preserve f.q.i. invariants.
- **Energy stability:** there exists $C_{i, \tau, \alpha}$ with $|\text{PE}_i^{\leq \tau}(F) - \text{PE}_i^{\leq \tau}(\tilde{F})| \leq C_{i, \tau, \alpha} \varepsilon^{\min\{1, \alpha\}}$ whenever $d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(F), \mathbf{T}_\tau \mathbf{P}_i(\tilde{F})) \leq \varepsilon$.
- **Spectral stabilization:** compute spectra on $L(C_\tau F)$; keep $(\beta, M(\tau), t)$ fixed. Averaging over N runs reduces variance as $N^{-1/2}$. Ignore aux-bar lifetimes ≤ 2 frames if declared in the manifest.
- **Resolution rule:** minimal resolvable feature length ≥ 3 grid steps; sweep τ on a lattice $\Delta\tau \leq \frac{1}{2}$ of the minimal resolvable bar length.
- **δ -ledger:** decompose per-step $\delta(i, \tau) = \delta_{\text{alg}}(i, \tau) \otimes \delta_{\text{disc}}(i, \tau) \otimes \delta_{\text{meas}}(i, \tau)$ in V , and aggregate as $\delta_{\text{tot}} = \bigotimes \delta_U$. (For the default Lawvere quantale, $\otimes = +$.); with layered boxes $(\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}})$ for cross-layer runs.

11.10. Saturation gate [Spec]

Declaration 10.17 (Window-local saturation). Fix $\tau^* > 0$ and parameters $\eta, \delta > 0$. On $[0, \tau^*]$, assume: (i) eventually the maximal *finite* bar length in $\mathbf{T}_\tau^* \mathbf{P}_i(F_t)$ is $\leq \eta$; (ii) eventually $d_{\text{int}}(\mathbf{T}_\tau^* \mathbf{P}_i(F_t), \mathbf{T}_\tau^* \mathbf{P}_i(F_{t'})) \leq \eta$; (iii) the edge gap $\delta = \tau^* - \max\{b_r < \tau^*\}$ satisfies $\delta > \eta$. Then, **within this window only**, adopt the temporary binary policy

Decision rule (Spec): $E_1(W) = 0$ or $\text{Ext}^1(\mathcal{R}(C_{\tau^*} F), k) = 0 \implies$ treat the PH-check as passed on this window.

11.11. Artifacts, manifests, and minimal schema

Remark 10.18 (Implementation notes). *Artifacts.* (i) bars.json/h5: records $\langle i, b_\ell, b_r, w \rangle$; (ii) spec.json/h5: positive eigenvalues of $L(C_\tau F)$ per degree; (iii) aux.json/h5: occupancies $E_r(j)$ and bin metadata; (iv) ext.json: boolean for $\text{Ext}^1(\mathcal{R}(C_\tau F), Q)$ with minimal witness; (v) phi.json: ranks of $\phi_{i,\tau}$ and $(\mu_{i,\tau}, u_{i,\tau})$. *Run log.* Store: sweep $\tau_{\min}:\Delta\tau:\tau_{\max}$; $(\alpha, w_i; \beta, M(\tau), t)$; discretization (grid/complex, step sizes); seeds; software versions; δ -ledger per step; bin window $[a, b]$, width β , under/overflow; mandatory spectral fields order=ascending, norm $\in \{\text{op}, \text{fro}\}$; overlap checks; Čech–Ext¹ status; tail-isomorphism flag; optional length-spectrum summary.

Minimal run.yaml block (augmented):

```
quantale:
  name: "[0,inf]_plus"
  op: "+"
  unit: 0.0
  order: "<="
layered_delta: {deltaGal: 0.002, deltaTr: 0.003, deltaFun: 0.002}
windows:
  domain: [[0,1), [1,2), [2,3)]
  collapse_tau: 0.08
  spectral_bins: {a: 0.0, beta: 0.02, bins: 96, boundary: "right-open"}
coverage_check: {length_sum: 3.0, length_target: 3.0, events_sum_equals_global: true}
overlap_checks: {local_equal_after_collapse: true, cech_ext1_ok: true, stability_band_ok: true}
spectral_policy: {order: "ascending", norm: "op"}
operations:
  - U: mollify; type: deletion; tau: 0.08; delta: {alg:0.004, disc:0.003, meas:0.001}
  - U: timestep; type: epsilon; tau: 0.08; eps:0.006; delta: {alg:0.000, disc:0.002, meas:0.001}
persistence:
  E1_zero_window: true
  PH1_zero: true
  Ext1_zero: true
  mu: 0
  nu: 0
  phi_iso_tail: true
length_spectrum: {degree: 1, tau: 0.08, eigenvalues: [0.24, 0.51, 0.78]}
spectral:
  ST_beta: 2
  ST_M_of_tau: "floor(0.5 * tau^1.5)"
  HT_t: [0.5*tau^-2, 1.0*tau^-2]
  auxBars_remaining: 0
budget: {sum_delta: 0.011, safety_margin: 0.025}
gate: {accept: true}
```

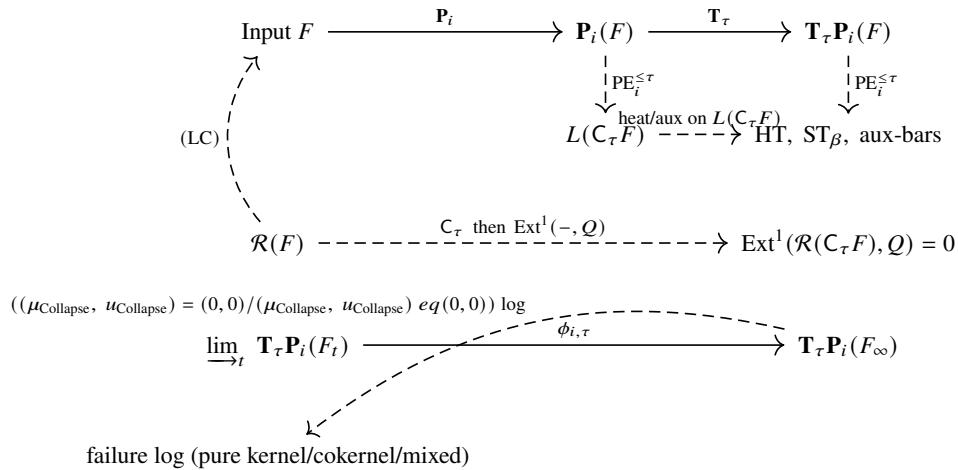
11.12. Compliance checks and unit tests

Declaration 10.19 (Minimal test suite). Every deployment must pass:

- **Stability test.** Under synthetic ε -perturbations, verify non-expansiveness of $\text{PE}_i^{\leq \tau}$ and stability of spectral indicators and aux-bars under the fixed policy.

- **Monotone-update test.** For a deletion-type update, confirm non-increase of $\text{PE}_i^{\leq \tau}$, spectral tails, and active-bin mass; record $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at fixed τ .
- **Cone-extension test.** Verify that $\phi_{i,\tau}$ is an isomorphism on a model tower (hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$) and that Type IV is excluded at that τ .
- **Categorical check.** On a curated sample, confirm stability of $\text{Ext}^1(\mathcal{R}(C_\tau F), Q) = 0$ under admissible f.q.i.-updates.
- **T- Λ_{len} -Consistency (optional).** Check that $\|\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F_t); [0, \tau]) - \Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F_{t'}); [0, \tau])\|_1 \leq \delta_{\text{win}}$ along declared stability bands (Def. 10.5), and that $\text{PE}_i^{\leq \tau}(F_t) = \text{trace}(\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F_t); [0, \tau]))$ holds to numerical tolerance.

11.13. Diagram (pipeline and logs)



11.14. Completion note

Remark 10.20 (No further supplementation required). This chapter fully integrates: (i) E_1 as first-class determinant on definable windows with finite-event/finite-Čech guarantees (App. H/J) and a reprise of the local bridge (Thm. 10.2); (ii) default Λ_{len} audit for stability-band diagnostics with invariance via Prop. 10.4; (iii) spectral auxiliaries restricted to deletion-type non-increase and ε -stability with proofs deferred to App. E; (iv) Overlap Gate enforcement and the unique comparison order; (v) tower diagnostics $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ and the joint monitoring protocol; (vi) noise/discretization rules and a quantale δ -ledger (with layered boxes); (vii) artifacts/manifests with mandatory audit fields; (viii) a minimal, IMRN/AiM-ready test suite including the optional T- Λ_{len} -Consistency check. All claims remain in the B-side single-layer scope; no additional supplementation is needed for operational use.

11.15. Guard-rails

Remark 10.21 (Scope and non-claims). This chapter specifies measurement protocols and auxiliaries at the persistence/spectral/categorical layers. No analytic regularity, group trivialization, or number-theoretic identity is asserted. All statements respect the guard-rails of Part I; in particular, no claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made, and μ_{Collapse} differs from the classical Iwasawa μ .

11 Chapter 12: Formal Test Suite and Open Problems

Badge policy.

- **[Prop]**: mathematics proved in Part I (core results; cite exact source).
- **[Thm]**: same as **[Prop]** but stated at theorem-level (cite exact source).
- **[Declaration]**: programmatic specification in the implementable range, verifiable by the test suite in this chapter.
- **[Conjecture]**: forward-looking statement; no claim beyond the stated scope.

12.0. Notation & conventions

- **Constructible range.** We identify $\text{Pers}_k^{\text{cons}}$ with the constructible subcategory of $[\mathbb{R}, \text{Vect}_k]$ and use $\text{Pers}_k^{\text{cons}}$ uniformly.
- **Truncation phrase.** “after applying \mathbf{T}_τ ; equivalently on $C_\tau F$ ” indicates computation at the persistence layer after truncation (hence equivalently on the filtered lift $C_\tau F$).
- **Comparison order (unique).** All PF/BC and metric/spectral comparisons use the *single* order

$$\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare in } \text{Pers}_k^{\text{cons}},$$

i.e. **T–PFBC–AfterCollapse**. Pre-collapse comparisons are out of scope (Chs. 1, 9, 11).

- **Generic–fiber dimension.** For a comparison map $\phi_{i,\tau}$ at fixed τ , \dim_k denotes the *generic–fiber dimension after truncation*, i.e. the multiplicity of $I[0, \infty)$ summands in $\mathbf{T}_\tau \mathbf{P}_i(-)$; informally, the $t \rightarrow \infty$ stable rank within the τ –window.
- **Spectral ordering and norms.** Positive eigenvalues of $L(C_\tau F)$ are listed in ascending order $\lambda_1 \leq \lambda_2 \leq \dots$. Matrix/operator norms are $\|\cdot\|_{\text{op}}$ and $\|\cdot\|_{\text{fro}}$; each test declares and logs its choice.
- **Obstruction totals and macros.** $\mu_{i,\tau} = \dim_k \ker \phi_{i,\tau}$, $u_{i,\tau} = \dim_k \text{coker } \phi_{i,\tau}$, totals $\mu_{\text{Collapse}} = \sum_i \mu_{i,\tau}$, $u_{\text{Collapse}} = \sum_i u_{i,\tau}$ (finite by bounded degree).
- **Endpoints.** Endpoint conventions and infinite bars follow the global policy (Appendix A); infinite bars are not removed by \mathbf{T}_τ and are clipped by the window in all windowed quantities.
- **Non-expansiveness.** We use the spelling “non-expansive”/“non-expansiveness” uniformly.
- **Quantale & V-enrichment.** All δ -budgets live in a fixed commutative quantale V (Appendix S); V -Lipschitz and V -nucleus properties are tested in T16. Layered boxes $(\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}})$ are mandatory (Ch. 9).
- **Definable windows.** Windows W are definable in a fixed o-minimal (Archimedean) or Denef–Pas (non-Archimedean) structure; definability and finite Čech depth are tested in T17.
- **Quantale convention.** We write \otimes for the quantale operation (YAML op); in the default Lawvere case, $\otimes = +$.

12.1. Badge inventory (representative items)

Badge	Representative items (label / location)
[Prop]	Stability, idempotence, and exactness of \mathbf{T}_τ (Prop. 2.5, Ch. 2); Shift–commutation / 1-Lipschitz for \mathbf{T}_τ (Lemma 2.4, Ch. 2); Operational coreflection C_τ^{comb} on the implementable range (Prop. 4.2, Ch. 5); Tower diagnosis: $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0))$ via cone extension; isomorphism criterion excluding Type IV (Prop. G.10, Appendix J).
[Thm]	One-way bridge: $\text{PH}_1(F) = 0 \Rightarrow \text{Ext}^1(\mathcal{R}(F), k) = 0$ under (B1)–(B3) (Thm. 2.27, Ch. 3); Local bridge on definable windows (Thm. 2.30, Ch. 3; reprise Ch. 11).
[Declaration]	Ch. 2: (co)limit and pullback compatibility <i>at the persistence layer only</i> (after \mathbf{T}_τ); Ch. 6: filtered-colimit stability in geometry; joint indicators and protocol (after truncation); Ch. 7: arithmetic tower stability; non-identity of μ_{Collapse} with Iwasawa μ ; Ch. 8: tropical shortening \Rightarrow weak group collapse; mirror transfer <i>non-expansive after truncation</i> ; Ch. 9: three-layer (Gal \rightarrow Trans \rightarrow Funct) compatibility as isomorphisms in $\text{Pers}_k^{\text{cons}}$ <i>after $\mathbf{T}_\tau \mathbf{P}_i$</i> ; Ch. 10: persistence-guided regularization; AK–NS hypothesis (programmatic); Ch. 11: joint monitoring, noise/discretization policy, minimal test suite; Saturation gate.
[Conjecture]	Cross-domain collapse propagation (Chs. 6–10); AK–NS (Ch. 10); mirror-side propagation (Ch. 8); Functorial transfer stability (Ch. 9).

12.2. Formal test suite (unit / integration / regression)

All tests operate at the truncated persistence, spectral (on $L(C_\tau F)$), and categorical layers and are f.q.i. invariant at the persistence layer. A test *passes* iff all stated pass-criteria are met and logs are complete. Pass-criteria must state whether indicators are evaluated *per degree* or *aggregated* across degrees; the choice must be fixed and logged for the run. Spectra use ascending order $\lambda_1 \leq \lambda_2 \leq \dots$, and the chosen norm $\|\cdot\|_{\text{op}}$ or $\|\cdot\|_{\text{fro}}$ must be declared and logged.

(T1) Stability under non-expansive updates [Unit]. *Input:* pairs $F \rightarrow F'$ with $d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(F')) \leq \varepsilon$.
Assertions: $|\text{PE}_i^{\leq \tau}(F) - \text{PE}_i^{\leq \tau}(F')| \leq C_{i,\tau,\alpha} \varepsilon^{\min\{1,\alpha\}}$ (after \mathbf{T}_τ ; equivalently on C_τ); spectra of $L(C_\tau F)$ vs. $L(C_\tau F')$ satisfy the fixed $(\beta, M(\tau), t)$ -policy stability bounds in the declared norm; $\text{Ext}^1(\mathcal{R}(C_\tau -), Q)$ is stable under admissible f.q.i. updates ($Q \in \{k[0]\}$).
Artifacts: bars.json, spec.json, ext.json; run.yaml (norm and spectral policy recorded).

(T2) Monotone update (deletion–/inclusion–type) [Unit]. *Input:* $F \rightarrow F'$ monotone.
Assertions: *Deletion–type:* $\text{PE}_i^{\leq \tau}$ and spectral indicators are non-increasing (after \mathbf{T}_τ). In the synthetic two-term cone test case satisfying the Ch. 4 cone-extension hypotheses, the map $\phi_{i,\tau}$ is an isomorphism, hence

$(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at fixed τ . *Inclusion-type*: stability only.

Artifacts: bars.json, spec.json, ext.json, phi.json; run.yaml.

(T3) Filtered-colimit stability [Integration]. *Input*: tower $\{F_\lambda\}_\lambda$.

Assertions: for fixed τ , $\phi_{i,\tau} : \varinjlim_\lambda \mathbf{T}_\tau \mathbf{P}_i(F_\lambda) \xrightarrow{\cong} \mathbf{T}_\tau \mathbf{P}_i(F_{\lambda_*})$; thus $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at that scale. Terminal symbol consistency is logged.

Artifacts: phi.json, run.yaml.

(T4) Mirror/tropical pipeline [Integration]. *Input*: X , tropical flow Trop_λ , realization F_λ , mirror functor Mirror .

Assertions: shortening factor $\kappa \leq 1$ implies non-increase of $\text{PE}_i^{\leq \tau}$ (after \mathbf{T}_τ); mirror transfer is non-expansive after truncation; group proxies (if used) meet weak-collapse thresholds (Ch. 8). Spectral eigenvalues ascending; norm declared.

Artifacts: per- λ bars/spec/ext/phi.json; run.yaml.

(T5) Three-layer compatibility [Integration]. *Input*: $\text{Gal} \rightarrow \text{Trans} \rightarrow \text{Func}$ data with comparison natural transformations (Ch. 9).

Assertions: after \mathbf{C}_τ and \mathbf{P}_i , commutativity holds up to isomorphism in $\text{Pers}_k^{\text{cons}}$ per degree; indicators consistent; failures typed and logged.

Artifacts: per-layer bars/spec/ext.json; global phi.json; run.yaml.

(T6) PDE monitoring loop [Regression]. *Input*: index set I (time/resolution/parameter), realization \mathcal{P} .

Assertions: Ch. 11 protocol holds; stable regime flags match logs of $\text{PE}^{\leq \tau}$, spectral indicators & aux-bars, Ext^1 , $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0))$; reporting choice (per degree vs. aggregated) fixed and logged; spectra ascending; norm declared.

Artifacts: bars/spec/aux/ext/phi.json over I ; run.yaml.

(T7) Saturation gate verification [Integration]. *Input*: window $[0, \tau^*]$ with candidate saturation (Ch. 11).

Assertions: verify saturation parameters (η, δ) and record the window-local **decision rule** enabled by the Saturation Gate: within $[0, \tau^*]$, if $E_1(W) = 0$ or $\text{Ext}^1(\mathcal{R}(\mathbf{C}_{\tau^*} F), k) = 0$, then the PH-check may be treated as passed on this window. (No logical equivalence is asserted.) *Artifacts*: bars.json, ext.json; run.yaml.

(T8) ε -clipping regression [Unit]. *Input*: paired runs with unclipped vs. ε -clipped $\mathbf{T}_\tau \mathbf{P}_i$ (Ch. 11).

Assertions: energy stability bound; $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0))$ computed on unclipped data and identical across the pair; logs distinguish clipping.

Artifacts: bars.json (both), phi.json; run.yaml.

(T9) MECE window coverage & event accounting [Unit]. *Input*: MECE windowing $\{[u_k, u_{k+1})\}_k$ and global range $[u_0, U)$.

Assertions: coverage equality; event counts add up (up to tolerance); uniform τ and bin policy unless justified and logged.

Artifacts: run.yaml (coverage_check), per-window bars.json.

(T10) A/B commutativity test for reflectors [Unit/Integration]. *Input*: two persistence-level reflectors

T_A, T_B , tolerance $\eta \geq 0$.

Assertions: $\Delta_{\text{comm}}(M; A, B) = d_{\text{int}}(T_A T_B M, T_B T_A M)$; pass if $\Delta_{\text{comm}} \leq \eta$, else fallback order is used and

defect added to δ^{alg} (Appendix L).

Artifacts: run.yaml (A/B policy), bars.json before/after.

(T11) Restart & Summability [Integration/Regression]. *Input:* windows with thresholds τ_k , budgets $\Sigma\delta_k(i)$, margins gap_{τ_k} .

Assertions: Restart: $\text{gap}_{\tau_{k+1}} \geq \kappa(\text{gap}_{\tau_k} - \Sigma\delta_k(i))$ (record κ). Summability: $\sum_k \Sigma\delta_k(i) < \infty$. Certificates paste.

Artifacts: run.yaml (restart/summability), gate logs, global certificate.

(T12) Trigger pack verification (domain-restricted) [Integration]. *Input:* declared triggers (e.g. PDE, Ch. 10).

Assertions: each trigger implies a B-Gate⁺ failure on the window; detection rate and false positives logged. Triggers are [Spec] and complement (not replace) B-Gate⁺.

Artifacts: run.yaml (thresholds), aux.json, phi.json, gate verdicts.

(T13) δ -ledger additivity & pipeline budget [Integration]. *Input:* steps U_m, \dots, U_1 with per-step collapses C_{τ_j} and bounds $\delta_j(i, \tau_j)$.

Assertions: verify

$$d_{\text{int}}(\dots) \preceq \bigotimes_{j=1}^m \delta_j(i, \tau_j), \quad \text{where } \otimes \text{ is the quantale operation (YAML: quantale.op).}$$

and that post-processing by 1-Lipschitz maps does not increase the bound (Appendix L).

Artifacts: run.yaml (δ -ledger), bars.json, distance logs.

(T14) Overlap Gate gluing test [Integration]. *Input:* charts X_1, X_2 with windows W_1, W_2 and overlap X_{12} ; fixed τ ; reflectors T_A, T_B .

Assertions: after \mathbf{P}_i and \mathbf{T}_τ , verify post-collapse local equivalence on X_{12} within budget, Čech–Ext¹ vanishing, A/B soft-commuting or logged fallback; construct glued truncated object and confirm B-Gate⁺.

Artifacts: run.yaml (overlap/AB/ δ), per-chart/overlap bars/ext.json; global verdict.

(T15) Length spectrum audit [Unit]. *Input:* $\mathbf{T}_\tau \mathbf{P}_i(F)$ and $\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F); [0, \tau])$.

Assertions: eigenvalue multiset equals clipped bar-length multiset (up to permutation); L^1 -mass equals $\text{PE}_i^{\leq \tau}(F)$. Hash or canonical ordering recorded.

Artifacts: bars.json, Lambda_len.json (optional list/hash), run.yaml.

(T16) V-shift (Quantale) test [Unit]. *Input:* a fixed commutative quantale V with operation \oplus , unit e , order \preceq ; Lawvere V -distance and V -shift operators S^v ($v \in V$); truncation \mathbf{T}_τ .

Assertions: (i) V -Lipschitz: $d_V(\mathbf{T}_\tau S^v M, \mathbf{T}_\tau S^v N) \preceq d_V(\mathbf{T}_\tau M, \mathbf{T}_\tau N)$ for all v (Ch. 2, Lemma “V-shift”); (ii) commutation: $\mathbf{T}_\tau \circ S^v \cong S^v \circ \mathbf{T}_\tau$ at the persistence layer; (iii) composition: $S^{v_2} \circ S^{v_1} \cong S^{v_1 \oplus v_2}$ and quantitative defect (if any) is recorded in δ^{alg} .

Artifacts: run.yaml (quantale:{name, op, unit, order, mode}), bars.json before/after S^v , distance logs.

(T17) Definable coverage & Čech finiteness [Integration]. *Input:* a window cover by formulas $\{\varphi_\alpha(x)\}$ in a fixed o-minimal or Denef–Pas structure; right-open windows $W_\alpha = \{x : \varphi_\alpha(x)\}$.

Assertions: (i) definability check passes for each φ_α ; (ii) finite event count on each W_α (piecewise constant Betti integrals; Appendix H); (iii) finite Čech depth on the cover (Appendix J); (iv) Overlap Gate checks pass

with logged δ -budgets; (v) optional confirmation of Ch. 3 local bridge when $E_1(W_\alpha) = 0$.

Artifacts: run.yaml (definable: {structure, window_formulae}), per-window bars.json, overlap logs.

(T18) Iwasawa control \Rightarrow Overlap Gate [Integration]. *Input:* arithmetic tower (Ch. 7) with a Control theorem yielding finite kernel/cokernel on comparison maps; Overlap Gate configuration.

Assertions: (i) finite kernel/cokernel are absorbed into δ^{alg} as per policy; (ii) post-collapse comparison maps satisfy $\phi_{i,\tau}$ isomorphism on windows where the control bound holds, hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at fixed τ ; (iii) Overlap Gate passes with recorded δ ; (iv) explicit bounds control_finite_bounds are logged.

Artifacts: run.yaml (iwasawa: {tower_level, control_finite_bounds}), phi.json, gate logs.

12.2a. Mandatory named tests (aliases/additions)

The following *named* tests are **mandatory**. Each comes with a canonical alias into (T1)–(T18) for reporting.

(T-ExtZero \Rightarrow PHZero) [Integration]. *Scope:* Only on definable windows with $E_1(W) = 0$ (Thm. 2.30) or under the Saturation Gate (Ch. 11, Decl. 10.17).

Assertions: On definable windows with $E_1(W) = 0$, verify that both $\text{PH}_1(C_\tau F|_W) = 0$ and $\text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0$ hold (window-local, after collapse). Outside this scope, the test is marked inapplicable. (window-local). Outside this scope, the test is marked inapplicable.

Alias: T7 (saturation) and T17 (definable). *Artifacts:* as in T7/T17.

(T-Countable-Cover) [Integration]. *Input:* a countable MECE cover $W = \bigsqcup_{n \geq 1} W_n$ with local finiteness on compact subintervals.

Assertions: (i) coverage/equality checks of T9 extend to the countable case; (ii) Overlap Gate passes on each finite subcover; (iii) certificates paste by Restart/Summability (T11).

Alias: T9+T11+T14. *Artifacts:* as in those tests.

(T-Delta-Sum-Converges) [Regression]. *Assertions:* $\sum_k \Sigma \delta_k(i) < \infty$ with logged tail bounds; global certificate exists.

Alias: T11 (Summability). *Artifacts:* restart/summability block.

(T-Lipschitz-AfterCollapse) [Unit]. *Assertions:* (i) \mathbf{T}_τ is 1-Lipschitz: $d_{\text{int}}(\mathbf{T}_\tau M, \mathbf{T}_\tau N) \leq d_{\text{int}}(M, N)$; (ii) each declared update is 1-Lipschitz *after* \mathbf{T}_τ within recorded ε ; (iii) deletion-type steps achieve non-increase of $\text{PE}_i^{\leq \tau}$.

Alias: T1+T2. *Artifacts:* as in T1/T2.

(T-Exactness-Persistence) [Unit]. *Input:* short exact sequences in $\text{Pers}_k^{\text{cons}}$ (implementable range).

Assertions: \mathbf{T}_τ is exact on these sequences; induced maps on barcodes respect subquotients; equality verified after collapse.

Alias: supports [Prop] “exactness of \mathbf{T}_τ ” and audits via T10 (A/B) when multiple reflectors are present. *Artifacts:* before/after bars.json.

(T-Iwasawa-Alignment) [Integration]. *Assertions:* control-theorem comparators align with δ -ledger: finite kernel/cokernel absorbed into δ^{alg} ; Overlap Gate passes; $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ at fixed τ .

Alias: T18. *Artifacts:* as in T18.

(T–PFBC–AfterCollapse) [Unit/Integration, optional]. *Assertions:* PF/BC steps computed object-wise in t then compared only *after* \mathbf{T}_τ ; non-expansiveness verified post-truncation; any discretization/sampling residuals are charged to $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$.

Alias: extends T5/T13 with PF/BC flags. *Artifacts:* PF/BC comparator logs.

(T– Λ_{len}) [Unit, optional]. *Assertions:* identical to T15; provide canonical hash for eigenvalue multiset of $\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F); [0, \tau])$.

Alias: T15. *Artifacts:* Lambda_len.json.

12.3. Reproducibility and logs

Every run ships with a manifest run.yaml declaring: sweep $\tau_{\min} : \Delta\tau : \tau_{\max}$; spectral policy $(\beta, M(\tau), t)$; discretization (grid/complex, steps); seeds; software versions; tower index set and cone extension (including the terminal symbol); pass-criteria (per-degree vs. aggregated); norm choice $\|\cdot\|_{\text{op}}$ or $\|\cdot\|_{\text{fro}}$; A/B tolerance η ; Restart constants (κ) and Summability evidence; Overlap Gate status; file pointers to bars/spec/aux/ext/phi.json (optionally .h5). Persistence quantities are computed after \mathbf{T}_τ ; equivalently on $C_\tau F$; and remain invariant under f.q.i. at the persistence layer.

Declaration 11.1 (Schema extension and mandatory fields [Spec]). The following run.yaml fields are *mandatory* for auditability (synchronized with Appendix G):

- **Quantale block** quantale:{name, op, unit, order, mode} (e.g. $[0, \infty)_+$, (max, +), probabilistic/product modes).
- **Layered δ** layered_delta:{deltaGal, deltaTr, deltaFun}.
- **Definable windows** definable:{structure, window_formulae} (structure $\in \{\mathbb{R}_{\text{an}}, \text{exp}, \text{DeneffPas}\}$).
- **Iwasawa** iwasawa:{tower_level, control_finite_bounds}.
- **AWFS/2-cell** awfs:{enabled: bool, two_cell_bounds: value}.
- **Overlap Gate** overlap_checks:{local_equiv, cech_ext1_ok, stability_band_ok}.
- **Length spectrum** Lambda_len per degree on $[0, \tau]$ (list or hash) for T15.
- **Spectral policy** spectral_policy:{order: "ascending", norm: "op"|"fro"}, and spectral_bounds:{lambda_min, lambda_max, lip_tol?}.
- **Persistence** explicit (μ, u) totals and tail-isomorphism flag phi_iso_tail.
- **Budget** sum_delta, safety_margin; for Restart, per-window gap_tau.
- **A/B** ab_test:{eta, policy, fallback}.

Remark 11.2 (Audit checklist). (i) Constructibility verified; (ii) Coefficient field fixed (Novikov allowed at [Spec]); (iii) Deletion– vs. inclusion–type correctly labeled; (iv) Uniform interleaving shifts ε_n bounded; (v) Same window for PE, spectral, aux-bars, Ext¹, and (μ, u) after \mathbf{T}_τ ; (vi) LC order: C_τ then \mathcal{R} (one-way bridge only); (vii) PF/BC prechecks for derived transfers (App. N); non-expansiveness only *after* truncation; (viii) Spectra ascending; norm declared; terminal symbol consistent; (ix) MECE coverage and event accounting satisfied (finite or countable); (x) A/B commutativity configured (T10) and logged; (xi) Restart/Summability evidenced (T11/T–Delta–Sum–Converges); (xii) Overlap Gate fields complete (T14); (xiii) Length spectrum audit recorded (T15/T– Λ_{len}); (xiv) Quantale/definable/Iwasawa/AWFS blocks complete (T16–T18); (xv) T–PFBC–AfterCollapse flags present when PF/BC is used.

Manifest template (YAML).

```
coeff_field: "k"          # or "Novikov(q)" [Spec-level]
tau_window: [0.05, 1.0]   # start, end
tau_step: 0.05
quantale:
  name: "[0,inf]_plus"
  op: "+"
  unit: 0.0
  order: "<="
  mode: "standard"        # or "probabilistic", "product"
layered_delta: {deltaGal: 0.002, deltaTr: 0.003, deltaFun: 0.002}
definable:
  structure: "R_an,exp"    # or "Denef-Pas"
  window_formulae:
    - "u <= t < u'"
    - "t in union_{j=1..m} (a_j, b_j]"
iwasawa:
  tower_level: 5
  control_finite_bounds: {kernel_le: 2, cokernel_le: 3}
awfs:
  enabled: true
  two_cell_bounds: 0.01
spectral:
  tail_beta: 2
  tail_cutoff_M_of_tau: "floor(0.5 * tau^1.5)"
  heat_t: [0.5*tau^-2, 1.0*tau^-2]
  aux_bins: {a: 0.0, beta: 0.02, bins: 96, boundary: "right-open"}
spectral_policy:
  order: "ascending"
  norm: "op"
spectral_bounds:
  lambda_min: 1.0e-6
  lambda_max: 10.0
  lip_tol: 0.02
tower:
  eps_interleave_max: 0.02
  terminal_symbol: "infty" # or "lambda_star"
  cone_extension: true
ab_test:
  eta: 0.01
  policy: "soft-commuting" # or "fallback:A_then_B"
restart_summability:
  kappa_min: 0.8
  sum_delta_bound: 0.05
windows:
  domain: [[0,1), [1,2), [2,3)] # finite or countable MECE cover
  collapse_tau: 0.08
coverage_check:
  length_sum: 3.0
  length_target: 3.0
  events_sum_equals_global: true
overlap_checks:
  local_equiv: true
```

```

    cech_ext1_ok: true
    stability_band_ok: true
pfb:
    policy: "after_collapse"      # T-PFBC-AfterCollapse
    residual_ledger: ["disc", "meas"]
persistence:
    PH1_zero: true
    Ext1_zero: true
    mu: 0
    nu: 0
    phi_iso_tail: true
Lambda_len:
    degree: 1
    tau: 0.08
    audit: "hash:2f4c...d1"
record:
    bars: true
    PE: {report: "per-degree", clipping: "epsilon=0.02"}
    aux: {lifetime_min_frames: 3}
    heat_trace: {ordering: "ascending", norm: "op"}
    ext1: true
    mu_nu: true
budget:
    sum_delta: 0.011
    safety_margin: 0.025
    gap_tau: 0.03
gate:
    accept: true
notes: "deletion-type only for monotonicity; LC with Rfun after truncation; PF/BC verified."

```

12.4. Open problems (selected)

Remark 11.3 (Open problems).

1. **Quantitative bridge.** Domain-wise sufficient conditions implying $\text{Ext}^1 = 0$ from decay of $\|\mathbf{CE}^{\leq \tau}\|_1$ and $\text{ST}_\beta^{\geq M(\tau)}$ (window-local).
2. **Colimit criteria.** Sharp hypotheses guaranteeing $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ beyond objectwise degreewise colimits.
3. **Failure lattice.** Finer invariants separating pure/mixed failures and anticipating Type IV at nearby scales.
4. **Spectral–persistence calibration.** Robust bounds between collapse energy and spectral tails under noise and discretization.
5. **Weak group collapse.** Persistence-level proxies vs. algebraic invariants without leaving the implementable range.
6. **Arithmetic towers.** Templates linking collapse diagnostics to Selmer/class growth while keeping $\mu_{\text{Collapse}} \leq q\mu_{\text{Iwasawa}}$.
7. **Langlands layers.** Minimal comparison data for truncated commutativity across $\text{Gal} \rightarrow \text{Trans} \rightarrow \text{Funct}$.

8. **PDE program.** Conditions under which persistence-guided regularization predicts classical regimes programmatically.
9. **Universality of T_τ .** Characterization of T_τ as Serre localization in the implementable range.

12.5. Final guard-rails

Remark 11.4 (Scope and non-claims). All specifications are confined to the persistence/spectral/categorical layers in the implementable range and are verifiable by the test suite above. No number-theoretic identity, analytic regularity theorem, or group trivialization is asserted. In particular, no claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made; only the one-way implication under (B1)–(B3) is used. The obstruction μ_{Collapse} is a collapse diagnostic and differs from the classical Iwasawa μ .

12.6. Effect and auditability

Remark 11.5 (Effect of the extensions). Relative to v16.0, the harness is strengthened by: (i) Overlap Gate gluing (T14) with post-collapse local equivalence and A/B soft-commuting; (ii) A/B tests with manifest-level tolerance (T10); (iii) Restart/Summability quantification (T11); (iv) Saturation Gate anchoring (T7); *and additionally* (v) V -shift verification for quantale-enriched runs (T16); (vi) definable coverage & Čech finiteness checks (T17); (vii) Iwasawa control integration with δ^{alg} absorption (T18); (viii) **mandatory** named tests $T\text{--ExtZero} \Rightarrow \text{PHZero}$, $T\text{--Countable--Cover}$, $T\text{--Delta--Sum--Converges}$, $T\text{--Lipschitz--AfterCollapse}$, $T\text{--Exactness--Persistence}$, $T\text{--Iwasawa--Alignment}$; (ix) optional $T\text{--PFBC--AfterCollapse}$ and $T\text{--}\Lambda_{\text{len}}$. The schema (Decl. 11.1) mandates quantale/definable/Iwasawa/AWFS/PFBC blocks, improving third-party auditability.

12.7. Conclusion

This chapter consolidates a complete, testable interface: a precise badge policy, a uniform notation layer, and a formal test suite spanning stability, monotone updates, filtered-colimits, mirror/tropical flows, Langlands triples, PDE pipelines, Overlap Gate gluing, quantale-enriched shifts, definable coverage, and Iwasawa control. All persistence-layer quantities are computed after T_τ ; spectral indicators are normalized (eigenvalues ascending; norm declared); categorical checks are performed only in the one-way direction. Reproducibility is enforced by a single manifest with mandatory fields. The *implementable range* is thus executable and auditable, with conservative guard-rails and clear open directions.

12.8. Completion note

Remark 11.6 (No further supplementation required). This chapter fully integrates: (i) MECE (finite/countable) window tests and event accounting; (ii) A/B commutativity with tolerance and fallback; (iii) Restart/Summability verification; (iv) Trigger pack validation; (v) δ -ledger additivity; (vi) Overlap Gate gluing (T14); (vii) Length spectrum audit ($T15/T\text{--}\Lambda_{\text{len}}$); (viii) Quantale V -shift tests (T16); (ix) definable coverage/Čech finiteness (T17); (x) Iwasawa control \Rightarrow Overlap Gate ($T18/T\text{--Iwasawa--Alignment}$); (xi) PF/BC enforcement after collapse ($T\text{--PFBC--AfterCollapse}$); (xii) a manifest schema synchronized with Appendix G. All items are consistent with the v16.0 guard-rails and cross-reference the proven core; no additional supplementation is needed for operational use as a formal test suite.

12.9. Machine-readable badge & test index (for automated extraction)

This block is purely auxiliary for reproducibility tools and can be ignored in print.

```

badge_index:
  proof_labels: ["prop:stability","lem:Vshift","prop:operational-coreflection","J:prop:diagzero"]
  theorem_labels: ["thm:PH1-to-Ext1","thm:E1-local"]
  declaration_chapters: [2,6,7,8,9,10,11,12]
mandatory_tests:
  - name: "T-ExtZero->PHZero"      ; alias: ["T7","T17"] ; scope: "definable_or_saturation"
  - name: "T-Countable-Cover"      ; alias: ["T9","T11","T14"]
  - name: "T-Delta-Sum-Converges"  ; alias: ["T11"]
  - name: "T-Lipschitz-AfterCollapse"; alias: ["T1","T2"]
  - name: "T-Exactness-Persistence"; alias: []
  - name: "T-Iwasawa-Alignment"    ; alias: ["T18"]
optional_tests:
  - name: "T-PFBC-AfterCollapse"    ; alias: ["T5","T13"]
  - name: "T-Lambda_len"           ; alias: ["T15"]
pfbc_policy: "after_collapse"
spectral_policy: {order: "ascending", norm: "op"}

```

Chapter 13: The Map of Validity and Defect Potential

13.0. Overview and Motivation

Part I established the *Unified Collapse Contract (UCC)* as a rigorous auditor: given an input F and collapse threshold τ , the system certifies validity through the inequality

$$\|\Sigma\delta(x)\|_V < \text{Gap}_\tau,$$

This certificate is sufficient for verification but inadequate for *exploration*. To study global mathematical families—such as flows of PDEs, arithmetic families of elliptic curves, or geometric variations—we require a framework that provides directional information: a means to navigate the parameter space \mathcal{M} .

This chapter introduces such a mechanism. We reinterpret the δ -ledger as a **scalar potential** $\Phi : \mathcal{M} \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$, providing a navigable landscape over which AI agents—defined in Chapter 14—may perform gradient descent, local search, or dimensional lifting. Paired with a definable partition of \mathcal{M} into *Terrain Cells*, the framework evolves from a passive auditor into an active navigation system.

13.1. From Collapse Diagnosis to Navigation

Let \mathcal{M} denote the moduli space of admissible inputs. We assume throughout that \mathcal{M} admits a stratification by definable sets (Appendix Q).

Definition 11.7 (Navigation Mode). In *Navigation Mode*, the AK pipeline does not reject an input $x \in \mathcal{M}$ when $\Sigma\delta(x) \geq \text{gap}_\tau$. Instead, it returns both the **magnitude** of the defect and, when available, an approximate gradient direction indicating how to reduce the defect within \mathcal{M} .

Specification 11.8 (Gradient Oracle (canon) [Spec]). In *Navigation Mode*, any reference to a “gradient” $\nabla\Phi$ means a *logged, reproducible gradient estimate* $\widehat{\nabla}\Phi$ produced by the following oracle GradEst.

Oracle interface. Given a state x in the navigation/search space, a scalar potential $\Phi(x)$ (Def. 11.10), and a policy block grad_policy recorded in run.yaml (Appendix G), the oracle returns:

$$\text{GradEst}(x; \text{grad_policy}) \rightsquigarrow (\widehat{g}, \widehat{\sigma}^2, \text{diag}),$$

where \hat{g} is the gradient estimate, $\hat{\sigma}^2$ is an estimated variance (or an upper bound), and `diag` is a structured diagnostic record (below).

Allowed methods (enumerated). `grad_policy.method` is one of:

`finite_difference` | `SPSA` | `surrogate`.

For `finite_difference`, `grad_policy.stencil` fixes the stencil (one-/two-sided, coordinate subset, step size ε). For `SPSA`, `grad_policy.seed` and perturbation distribution are fixed. For `surrogate`, the surrogate family and fitting seed are fixed.

Norms and step semantics. All step decisions use the norm recorded in `grad_policy.norm` and must be consistent with the spectral/persistence norms declared in `spectral_policy / pipeline.metric` (Appendix G). Any smoothing/regularization used by GradEst must be declared in `grad_policy.regularization`.

Audit and δ -ledger charging. Gradient estimation introduces an approximation component that must be charged into `operations[*].delta.sources` and aggregated into `budget.sum_delta` via the manifest quantale (Ch. 12, Dec. 11.1). At minimum, the diagnostic record `diag` contains:

- `method`, `stencil` (if applicable), `seed`,
- `eval_count`, `eps` (if applicable),
- `variance` (or certified bound), and
- `delta_charge` (the amount charged for this oracle call).

The corresponding per-step action log entry is mandatory (Appendix U).

Core/Search separation guard. No $\nabla\Phi$ claim is admissible unless it is backed by a GradEst record in `run.yaml` and Appendix U for the same `run_id`.

[Spec] Gradient output. When available, the pipeline returns an approximate descent direction for the Defect Potential Φ_τ (see Remark 11.11).

Thus the goal is no longer merely to certify a point but to identify the region:

$$Z_{\text{valid}} := \Phi^{-1}([0, \text{gap}_\tau]).$$

13.2. The Defect Potential $\Phi(x)$

The δ -ledger encodes algebraic, numerical, and functorial deviations from ideal collapse. To combine these into a single invariant, we apply a monotone scalarization.

Definition 11.9 (Scalarization). Let V be the Quantale of δ -budgets. A map $\|\cdot\|_V : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a *scalarization* if:

1. $\|0_V\|_V = 0$;
2. $\delta_1 \preceq \delta_2 \Rightarrow \|\delta_1\|_V \leq \|\delta_2\|_V$;
3. $\|\delta_{\text{alg}} \oplus \delta_{\text{disc}}\|_V \geq \|\delta_{\text{alg}}\|_V$ (algebraic defects contribute persistently).

Typical examples include ℓ^1 or ℓ^∞ norms when $V = [0, \infty]^k$.

Definition 11.10 (Defect Potential Φ_τ). Let $x \in \mathcal{M}$, and let $\Sigma\delta(x)$ denote the δ -budget *after* applying collapse T_τ . Let $(\mu(x), u(x))$ be the tower obstruction indices (Chapter 4). The *Defect Potential* is:

$$\Phi_\tau(x) := \|\Sigma\delta(x)\|_V + \lambda_{\text{sing}} \mathcal{I}_{\text{IV}}(x),$$

where $\lambda_{\text{sing}} \gg 1$ and

$$\mathcal{I}_{\text{IV}}(x) = \begin{cases} 1 & \text{if } (\mu(x), u(x)) \text{ eq}(0, 0), \\ 0 & \text{otherwise.} \end{cases}$$

Remark 11.11 (Gradient estimation and logging [Spec]). On each Terrain Cell, an approximate descent direction is computed for Φ_τ using a declared policy `grad_policy`: (i) finite differences, (ii) random perturbations, or (iii) a surrogate model. The Lipschitz metric on \mathcal{M} is evaluated in the declared norm $\|\cdot\|_{\mathcal{M}}$ (e.g. ℓ^2 or ℓ^∞) and logged. Any estimation/sampling residual is charged to $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$ and recorded in the δ -ledger.

Remark 11.12 (Geometric Stratification and Lifting Trigger). The potential Φ_τ stratifies the parameter space \mathcal{M} into three operational regimes, each prescribing a distinct AI-agent behavior:

- **Plain of Truth** ($\Phi(x) < \text{gap}_\tau$). Collapse is certified by UCC. The agent *records the region as valid* and explores boundaries.
- **Ridge of Noise** ($\text{gap}_\tau \leq \Phi(x) < \lambda_{\text{sing}}$). Only Types I–III defects occur. The agent performs **local gradient descent** (or random-walk refinement) within the Terrain Cell to search for a descending path.
- **Peak of Singularity** ($\Phi(x) \geq \lambda_{\text{sing}}$). Indicates essential Type IV obstruction. The agent must trigger a **Dimensional Lifting** request (Chapter 14), adding auxiliary axes to escape the singular fiber.

This stratification gives Φ an operational semantics and links it directly to the autonomous behaviors of Chapter 14.

13.3. Terrain Cells and Definable Geometry

Optimization on \mathcal{M} requires discretization compatible with definability and uniformity.

Definition 11.13 (Terrain Cell). A *Terrain Cell* $W_\alpha \subset \mathcal{M}$ is a definable, bounded subset satisfying:

1. **Uniform Constructibility:** The persistence diagrams $\mathbf{P}_i(F_x)$ admit uniform constructible description for all $x \in W_\alpha$.
2. **Lipschitz Budget:** The assignment $x \mapsto \Sigma\delta(x)$ is Lipschitz on W_α .
3. **MECE Partition:** The family $\{W_\alpha\}$ is mutually exclusive and collectively exhaustive.

Declaration 11.14 (Local convexity proxy [Spec]). Away from Type IV regions, Φ_τ is treated as approximately convex on sufficiently small Terrain Cells for the purpose of local search.

13.4. Structural Regularity (programmatic)

Declaration 11.15 (AK Global Validity via bounded potential [Spec]). Let \mathcal{M} be a path-connected definable component. Assume (1)–(3). Then, for every $x \in \mathcal{M}$, the UCC checks pass in the implementable range after truncation: B-Gate⁺ holds at the declared τ , and the local certificates glue (across the declared cover) to a global post-collapse certificate in $\text{Pers}_k^{\text{cons}}$ (up to the declared δ -ledger budgets).

Conjecture 11.1 (Interpretation under domain hypotheses). *Under the AK–NS hypothesis (Ch. 10) (resp. arithmetic realization hypotheses in Ch. 7), a globally bounded defect potential suggests absence of singular behavior (resp. alignment of ranks) in the monitored regime.*

Remark 11.16 ([Spec] Rationale (replacing the proof environment)). This paragraph records the intended implication chain of the programmatic declaration 11.15 under the stated hypotheses. Bounded potential is used as a proxy for uniform feasibility of B-Gate⁺ (with declared margins), the Type IV veto removes invisible obstructions, and definable coverage together with Restart/Summability is the gluing mechanism for local certificates. This is not a completed mathematical proof; it is an auditable design rationale for HDPS navigation.

13.5. Summary

This chapter transforms AK-HDPST from a passive diagnostic mechanism into a geometric navigation framework. The Defect Potential Φ supplies a scalar field governing movement, and Terrain Cells provide local uniformity for optimization. Chapter 14 introduces the autonomous agents—Hunter, Mapper, Lifter—that operate over this landscape.

Chapter 14: AI Agents — Hunter, Mapper, and Lifter

14.0. Overview and Agent Taxonomy

The scalar field Φ (Chapter 13) transforms collapse diagnostics into a navigable landscape on the parameter space \mathcal{M} . To explore this landscape efficiently and safely, we introduce three autonomous but contract-bounded agents:

- **Hunter** — performs local optimization of Φ within Terrain Cells, discovering valid regions and local minima.
- **Mapper** — assembles validated Terrain Cells into a coherent global structure using Overlap Gates.
- **Lifter** — resolves essential singularities (Type IV points) through controlled dimensional extension.

All agents operate under the *Unified Collapse Contract (UCC)*: they may propose actions, but acceptance and certification is performed only by the AK Core.

14.1. The Hunter: Regime-Aware Search Strategy

Navigation Mode (Definition 13.1) assigns each point $x \in \mathcal{M}$ to one of the three regimes determined by $\Phi(x)$:

$$\text{Plain} \quad (\Phi < \text{gap}_\tau), \quad \text{Ridge} \quad (\text{gap}_\tau \leq \Phi < \lambda_{\text{sing}}), \quad \text{Peak} \quad (\Phi \geq \lambda_{\text{sing}}).$$

The Hunter realizes these semantics operationally.

Remark 11.17 (Gradient oracle and logging [Spec]). In this chapter, “ $\nabla\Phi(x)$ ” denotes an *estimated descent direction* $g(x)$ for the scalar potential Φ_τ (Ch. 13), computed under a declared `grad_policy` (finite differences / random perturbations / surrogate). The metric on \mathcal{M} is evaluated in the declared norm $\|\cdot\|_{\mathcal{M}}$. All estimation and sampling residuals are charged to $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$ and recorded in the δ -ledger. (See Remark 11.11 and the manifest schema in Ch. 12.)

Definition 11.18 (Hunter State). A Hunter maintains a state

$$S_k = (x_k, W_k, \Phi_\tau(x_k), g_k, \text{meta}_k),$$

where g_k is an estimated descent direction produced by the gradient oracle (Remark 11.17), and meta_k logs `grad_policy`, $\|\cdot\|_{\mathcal{M}}$, step size, and seeds.

Specification 11.19 (Hunter Protocol). At iteration k , the Hunter acts according to the regime at x_k :

1. **Plain Regime** ($\Phi(x_k) \leq \text{gap}_\tau$): **Verification.** Trigger the full B-Gate⁺ check. If verified, mark W_k as valid and initiate exploration of its boundary ∂W_k .
2. **Ridge Regime** ($\text{gap}_\tau < \Phi(x_k) < \lambda_{\text{sing}}$): **Descent.** Compute $g_k = \text{GradEst}(\Phi_\tau, x_k; \text{grad_policy})$ and propose

$$x_{k+1}^{\text{prop}} := x_k - \alpha_k g_k, \quad x_{k+1} := \Pi_{W_k}(x_{k+1}^{\text{prop}}),$$

where Π_{W_k} is the declared projection/clipping policy to keep the iterate inside the current Terrain Cell. Accept the step only if $\Phi_\tau(x_{k+1}) \leq \Phi_\tau(x_k) - \rho$ (margin $\rho \geq 0$ declared and logged); otherwise shrink α_k and retry up to a declared limit. If $\|g_k\|_{\mathcal{M}}$ falls below a declared threshold, apply a controlled random perturbation (seed logged). All estimation/projection residuals are charged to $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$.

3. **Peak Regime** ($\Phi(x_k) \geq \lambda_{\text{sing}}$): **Escalation.** Terminate local search and invoke the Lifter to escape essential obstructions.

Every action is recorded in the *Hunter Action Log* (Appendix U) to guarantee reproducibility.

14.2. The Mapper: Global Assembly of Certificates

The Mapper ensures that local certificates produced by Hunters combine into a globally coherent proof artifact.

Definition 11.20 (Coverage Graph). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph whose vertices are Terrain Cells marked valid. An edge $(\alpha, \beta) \in \mathcal{E}$ exists when the Overlap Gate (Chapter 5) passes on the intersection $W_\alpha \cap W_\beta$.

Specification 11.21 (Mapper Protocol). The Mapper performs the following loop:

1. **Ingest Validated Cells.** Receive validated W_α from Hunters.
2. **Execute Overlap Gates.** Verify consistency on each intersection $W_\alpha \cap W_\beta$.
3. **Update Coverage Graph.** Add edges or merge components accordingly.
4. **Check Global Coverage.** In bounded domains: check whether a connected component covers the target domain. In unbounded domains: verify asymptotic stability of certificates.

When a connected component covers the domain of interest, the Mapper issues a **Global Certificate**.

14.3. The Lifter: Controlled Dimensional Extension

The Lifter responds exclusively to essential singularities (Type IV points: $\mu, u \neq 0$). Unlike the Hunter, its role is not optimization but *geometric escape*.

Definition 11.22 (Dimensional Lifting). If \mathcal{M}_n is the base parameter space, a *lifting* is an embedding

$$\iota : \mathcal{M}_n \hookrightarrow \mathcal{M}_{n+k}$$

adding k auxiliary coordinates (e.g. smoothing width, spectral softness, auxiliary weights, or arithmetic depth). The object F is pulled back to a lifted object \tilde{F} .

Specification 11.23 (Lifter Protocol). Given a singular point x_{sing} :

1. **Select Axis Type.** Choose an auxiliary axis appropriate to the obstruction (e.g. spectral, geometric, arithmetic).
2. **Construct Lifted Neighborhood.** Build a definable neighborhood $\tilde{W} \subset \mathcal{M}_{n+1}$ around $(x_{\text{sing}}, 0)$.
3. **Spawn New Hunter.** Launch a new Hunter on \tilde{W} to seek descent directions in the enlarged space.

All lifting steps are subject to the Lifting Penalty (Section 14.4).

14.4. Safety: The Lifting Penalty δ^{lift}

To prevent trivial “solve-by-infinite-lifting”, dimensional lifting incurs a cost charged against the global δ -budget.

Definition 11.24 (Lifting Penalty). For a lifting depth k , the penalty is a monotone function

$$\delta^{\text{lift}}(k) \in V,$$

commonly exponential, e.g.

$$\delta^{\text{lift}}(k) = C_{\text{lift}} 2^k.$$

Definition 11.25 (Augmented Gap Constraint). Any lifted search path must satisfy the UCC constraint in the declared comparison mode: either in V -order (via an embedding of gap_τ into V), or after scalarization:

$$\|\Sigma\delta(x) \oplus \delta^{\text{lift}}(k(x))\|_V < \text{gap}_\tau.$$

The chosen mode is declared in `run.yaml`.

Remark 11.26 (Operational Meaning). The Lifter is not a universal escape mechanism. It is a *high-cost extension operator* whose use must be justified by essential obstructions. In practice, only a small number of lifts are admissible under the UCC budget.

[Spec] Hunter Action Log. Each iteration logs $(x_k, W_k, \Phi_\tau(x_k))$, `grad_policy`, $\|\cdot\|_{\mathcal{M}}$, g_k , α_k , projection events, accept/reject, seeds, and the charged $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$.

14.5. Summary

Hunter, Mapper, and Lifter jointly implement the active exploration engine of AK-HDPST. Hunter minimizes the Defect Potential Φ , Mapper assembles local certificates into a global proof, and Lifter resolves essential singularities via controlled dimensional extension. Together they provide a sound, reproducible, and bounded search mechanism for the validity landscape described in Chapter 13.

Chapter 15: Collapse-Based Optimization Protocols

15.0. Overview and Motivation

Chapter 14 defined the agents (Hunter, Mapper, Lifter) as the actors of the navigation system. This chapter specifies the *optimization scripts* they follow.

Unlike classical optimization where the cost function is smooth and explicit, the Defect Potential $\Phi_\tau(x)$ is derived from persistence diagnostics and quantale-valued error ledgers. It can be non-convex, piecewise-defined, and expensive to evaluate. We therefore require robust protocols combining:

- **Local Descent:** Exploit approximate gradients where Φ is locally regular (Ridge regime).
- **Restart Logic:** Use the Restart Lemma (Appendix J) to escape shallow local minima caused by window scale.
- **Lifting Heuristics:** Recognize when failure is due to essential obstructions rather than numerical noise.

[Spec] Logging. CBGD stencils/samples, norm $\|\cdot\|_{\mathcal{M}}$, grad_policy, accept/reject, restart events, and the charged $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$ are mandatory log fields.

15.1. Search Strategies and Restart Logic

Within a Terrain Cell W , the primary operation of a Hunter is to minimize Φ under the Ridge regime ($\text{gap}_\tau < \Phi < \lambda_{\text{sing}}$).

Remark 11.27 (Discrete gradient oracle, norm, and ledger **[Spec]**). In this chapter, “ $\nabla_\delta \Phi(x)$ ” denotes an *estimated* descent direction $g(x)$ produced by a declared gradient policy grad_policy (e.g. finite stencil / random perturbations / surrogate). All distances and Lipschitz checks on \mathcal{M} use a declared norm $\|\cdot\|_{\mathcal{M}}$. Any estimation/sampling/projection residuals introduced by the oracle are charged to $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$ and recorded in the Hunter Action Log and run.yaml. (Consistency with Remark 11.17 and Ch. 12 is mandatory.)

Definition 11.28 (Collapse-Based Gradient Descent (CBGD)). Let $x_k \in W \subset \mathcal{M}$. Compute a discrete descent direction $g_k = \text{GradEst}(\Phi_\tau, x_k; \text{grad_policy})$ using a declared stencil/sampling policy on W (Remark 11.27). If $\|g_k\|_{\mathcal{M}} = 0$ (below a declared threshold), use a controlled perturbation (seed logged) to obtain $g_k \neq 0$. Propose

$$x_{k+1}^{\text{prop}} := x_k - \alpha_k \frac{g_k}{\|g_k\|_{\mathcal{M}}}, \quad x_{k+1} := \Pi_W(x_{k+1}^{\text{prop}}),$$

where Π_W is the declared projection/clipping rule to keep iterates in W . All oracle/projection residuals are charged to $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$. The step size α_k is adaptive and governed by the acceptance rule in Spec. 11.29.

Specification 11.29 (Restart Logic via Convergence Manager). Fix declared tolerances $\rho \geq 0$ (descent margin), $N \in \mathbb{N}$ (stagnation horizon), and a τ -restart schedule τ_k on a declared lattice. At each step:

1. **Accept/Reject.** Accept x_{k+1} only if $\Phi_\tau(x_{k+1}) \leq \Phi_\tau(x_k) - \rho$. Otherwise shrink α_k and retry up to a declared limit; log retries.
2. **Stagnation.** If $\Phi_\tau(x_k) > \text{gap}_\tau$ and $|\Phi_\tau(x_{k+\ell}) - \Phi_\tau(x_k)| \leq \rho$ for $1 \leq \ell \leq N$, declare stagnation and perform a **Window Restart**: subdivide W into smaller Terrain Cells W' (MECE, policy logged), and rerun CBGD on each W' under the same grad_policy and $\|\cdot\|_{\mathcal{M}}$.

3. **τ -Restart (scale artifact check).** If stagnation persists, switch to the next τ_{k+1} in the declared schedule and recompute $\Phi_{\tau_{k+1}}$ and the UCC check $\text{Gap}_{\tau_{k+1}} > \Sigma\delta$. Record (τ_k, τ_{k+1}) and the incremental ledger consumption.
4. **Local Minimum (resolution-bounded).** If no restart (cell refinement nor τ -restart) yields a descending move, mark x_k as a *resolution-bounded local minimum* and hand off to the Lifter heuristic (Sec. 15.2).

All decisions must be recorded in the Hunter Action Log, including `grad_policy`, $\|\cdot\|_{\mathcal{M}}$, stencil size / samples, seeds, α_k , accept/reject, and the charged $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$.

Remark 11.30 (Quantale Awareness). CBGD minimizes the scalarized quantity $\|\Sigma\delta\|_V$, but the full quantale-valued vector δ is always retained in the Hunter Action Log. This allows post hoc analysis of which component (algebraic vs. numerical, commutation vs. truncation) dominates the defect at a local minimum.

15.2. Handling Local Minima via Lifting Heuristics

When a Hunter is trapped at a local minimum x^* satisfying $\Phi(x^*) \geq \text{gap}_\tau$, the system must decide whether this reflects:

- a genuine candidate counterexample, or
- a projection artifact caused by insufficient dimension.

Definition 11.31 (Lifting Condition). A local minimum x^* is a candidate for Dimensional Lifting if:

1. **Persistent Obstruction:** $\Phi(x^*)$ remains above gap_τ under a τ -sweep (varying collapse scale within admissible bounds).
2. **Type IV Signature:** The tower diagnostics at x^* satisfy $(\mu(x^*), u(x^*)) \text{ eq}(0, 0)$.
3. **Cost Feasibility:** The base Lifting Penalty obeys $\delta^{\text{lift}}(1) < \text{gap}_\tau$, so that a single lift is admissible under the UCC.

Specification 11.32 (Lifting Heuristic Protocol). Given a local minimum x^* satisfying the Lifting Condition:

1. **Freeze Local Search.** Suspend further CBGD updates at x^* .
2. **Propose Auxiliary Axis.** The Lifter proposes an axis \mathcal{A} in accordance with Appendix U (e.g. smoothing parameter, spectral softening, arithmetic depth).
3. **Test Lift Gradient.** Evaluate the one-sided directional derivative (or finite difference) of Φ along \mathcal{A} at $(x^*, 0)$ in the lifted space $\mathcal{M} \times \mathcal{A}$.
4. **Commit or Reject Lift.**
 - If a direction with $\partial_{\mathcal{A}}\Phi < 0$ exists and the augmented budget $\|\Sigma\delta \oplus \delta^{\text{lift}}(1)\|_V < \text{gap}_\tau$ holds, commit the lift and spawn a new Hunter on $\mathcal{M} \times \mathcal{A}$.
 - If all tested axes yield non-decreasing Φ , label x^* as a **Terminal Singularity** (hard instance under the current lifting policy) and halt lifting; no mathematical counterexample claim is made.

15.3. Summary

This chapter specifies the optimization core of the AK-HDPST search engine:

- **CBGD** provides a regime-aware gradient descent adapted to the structure of Φ .
- **Restart Logic** uses window refinement to distinguish spurious local minima from resolution artifacts.
- **Lifting Heuristics** decide when and how to escape topological traps by controlled dimensional extension, subject to the Lifting Penalty of Chapter 14.

Together with Chapters 13 and 14, these protocols endow the Hunter and Lifter with a mathematically disciplined behavior over the Defect Potential landscape.

Chapter 16: Bridge Programs and Spectral-Gap Windows

16.0. Overview and motivation (Search layer only)

Chapter 3 established the *one-way bridge*

$$\mathrm{PH}_1(C_\tau F) = 0 \implies \mathrm{Ext}^1(\mathcal{R}(C_\tau F), k) = 0 \quad \text{under (B1)–(B3).}$$

This direction is a **Core** result (Part I) and is used by the AK Core.

In contrast, many global programs (PDE families, arithmetic families, etc.) would like to use categorical evidence to guide topological decisions operationally. In particular, one may wish—at the *Search Layer only*—to treat

$$\mathrm{Ext}^1(\mathcal{R}(C_\tau F|_W), k) \approx 0$$

as evidence for $\mathrm{PH}_1(C_\tau F|_W) = 0$ on a window/cell W . This chapter defines **Bridge Programs** B1–B3 that govern such *reverse use* under strict guard-rails.

A-plan principle (scope-first). Reverse use $\mathrm{Ext}^1 \implies \mathrm{PH}_1$ is *never* permitted globally or generically. It is permitted only in the definable_or_saturation scope that is already audited by the mandatory named test $\mathrm{T}\text{--}\mathrm{ExtZero} \implies \mathrm{PHZero}$ (Ch. 12, §12.2a; alias: T7/T17). Spectral information may be used only as an *auxiliary* safety check and does *not* expand the scope.

16.0+. Standing conventions (inherit)

All comparisons obey the unique order:

$$\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare in } \mathrm{Pers}_k^{\mathrm{cons}},$$

i.e. **T–PFBC–AfterCollapse**. All quantities used in this chapter are computed *after* \mathbf{T}_τ (equivalently on $C_\tau F$). Spectral conventions follow Ch. 11/12: eigenvalues in ascending order; norm choice $\|\cdot\|_{\mathrm{op}}$ or $\|\cdot\|_{\mathrm{fro}}$ declared and logged.

16.1. Program B1: the reverse-use problem (Search-layer) [Spec]

Definition 11.33 (The reverse-use problem). Fix a Terrain Cell $W \subset \mathcal{M}$ and a collapse scale $\tau > 0$. The *reverse-use problem* asks for conditions under which the Search Layer is allowed to *treat* the operational implication

$$\mathrm{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0 \implies \mathrm{PH}_1(C_\tau F|_W) = 0$$

as a local proxy, without asserting any new Core equivalence.

Specification 11.34 (Bridge Program B1 (A-plan): reverse use under definable_or_saturation only). Bridge Program B1 issues a **Reverse Certificate** on a Terrain Cell W only if the following *Collapse-Consistent Conditions* (CCC) hold.

1. **Scope Gate (mandatory).** Program B1 is applicable only if either:

- (a) W is definable and $E_1(W) = 0$ (local bridge scope; Thm. 2.30), or
- (b) the Saturation Gate is active on W at τ^* (Decl. 10.17).

Otherwise Program B1 is inapplicable and *no reverse propagation is permitted*.

- 2. **UCC compliance and ordering.** All computations are performed *after* T_τ (equivalently on $C_\tau F$), and all comparisons use **T-PFBC-AfterCollapse**. Pre-collapse comparisons are out of scope.
- 3. **Tower stability.** The tower diagnostics vanish on W at the declared τ : $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ (Ch. 4).
- 4. **Potential bound (Search-layer safety).** The Defect Potential satisfies $\Phi_\tau(x) < \text{gap}_\tau$ for all $x \in W$ (Ch. 13). This ensures UCC-style budget safety at the Search Layer.
- 5. **Test gating (mandatory named test).** The mandatory named test $T\text{-ExtZero} \Rightarrow \text{PHZero}$ (Ch. 12, §12.2a; alias: T7/T17) is applicable and passes, with full logs and artifacts.

Certificate semantics (Search layer only). If CCC holds, B1 records the local statement

$$\text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0 \implies \text{PH}_1(C_\tau F|_W) = 0 \quad (\text{window-local, Search layer only}).$$

This does *not* assert any Core equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$; it is a gated proxy rule used only by the Search Layer.

Remark 11.35 (Failure modes and dispositions). If Program B1 is inapplicable (outside definable_or_saturation), the cell W remains undecided under reverse use. The Search Layer may (i) continue exploration using Φ and the Hunter protocol, (ii) request refinement/restart, or (iii) flag for lifting when Type IV is indicated.

16.2. Optional spectral safety: spectral-gap windows [Spec]

Positioning (A-plan). Spectral information is an *auxiliary safety check*. It can reduce false positives from near-zero numerical artifacts, but it *never expands the applicability* of B1. In particular, satisfying a spectral gap *cannot* make B1 applicable outside definable_or_saturation.

Definition 11.36 (Spectral gap at scale τ). Let $L_1(C_\tau F_x)$ denote the normalized combinatorial Hodge Laplacian on degree 1 for the truncated object $C_\tau F_x$ (Ch. 11). Let $\lambda_1(x) \leq \lambda_2(x) \leq \dots$ be its positive eigenvalues (ascending order). Define the *spectral gap*:

$$\gamma_\tau(x) := \lambda_1(x),$$

with the convention $\gamma_\tau(x) = +\infty$ if there is no positive spectrum.

Definition 11.37 (Noise floor from the ledger). Let $\Sigma\delta(x) \in V$ be the aggregated post-collapse budget (Appendix S/L), and let $\|\cdot\|_V$ be the fixed scalarization used in Ch. 13. Define the *noise floor* on a cell W by

$$\text{NF}(W) := \sup_{x \in W} \|\Sigma\delta(x)\|_V.$$

Optionally, if the implementation separates measurement/discretization components, record $\text{NF}_{\text{meas+disc}}(W) := \sup_{x \in W} \|\delta_{\text{meas}}(x) \oplus \delta_{\text{disc}}(x)\|_V$.

Definition 11.38 (Spectral-gap condition (auxiliary)). A Terrain Cell W satisfies the *Spectral-Gap Condition* at scale τ if there exists a constant $c > 1$ such that

$$\inf_{x \in W} \gamma_\tau(x) > c \cdot \text{NF}(W).$$

The policy fixes c (default $c = 2$) and logs it in `run.yaml`.

Specification 11.39 (Design guarantee: spectral robustness (auxiliary safety)). When Program B1 is applicable (Scope Gate satisfied) and CCC holds, the Spectral-Gap Condition provides an auxiliary guard against misclassifying a small positive mode as “zero” due to numerical noise. All spectral parameters must be fixed and logged:

- eigenvalue order: ascending;
- norm choice for matrix perturbations: op or fro;
- spectral bounds (at least $\lambda_{\min}, \lambda_{\max}$) and optional `lip_tol`;
- the constant c in Def. 11.38;
- the backend precision / solver tolerance used for eigen-computation.

Passing the spectral check may be recorded as `spectral_aux_ok: true` but *cannot* override any failure of the Scope Gate or the mandatory named test.

Remark 11.40 (Minimal log block). A minimal manifest extension (consistent with Ch. 12, Decl. 11.1) is:

```
reverse_bridge:
  enabled: true
  scope: "definable_or_saturation"
  mandatory_test: "T-ExtZero->PHZero"
spectral_aux:
  enabled: true
  c: 2.0
  gamma_inf_min: 0.15
  noise_floor_sup: 0.05
```

16.3. Program B2: Global assembly at the Search Layer [Spec]

Specification 11.41 (Program B2: Search-layer regularity via coverage). **Input:** a parameter space \mathcal{M} and a realization F (Part II).

Goal: produce a *Search-layer* coverage artifact indicating that the domain of interest is covered by locally validated cells, with all gluing checks passing.

Procedure:

1. **Decomposition.** Hunters propose a cover of \mathcal{M} by Terrain Cells $\{W_\alpha\}$ satisfying definability/MECE and Restart/Summability requirements (Ch. 12, T9/T11/T17).
2. **Local evaluation.** On each W_α , run UCC/B-Gate⁺ checks and diagnostics. If Program B1 is applicable and passes CCC (including $T\text{-ExtZero} \Rightarrow PHZero$), mark W_α as reverse-validated. If B1 is inapplicable, keep the cell undecided under reverse use.

3. **Singularity handling.** If the Peak regime is detected (Type IV indicated by $(\mu, u) \text{ eq}(0, 0)$), invoke the Lifter under the Lifting Penalty (Ch. 14) and record outcomes. Unresolved peaks are marked barrier.
4. **Gluing.** The Mapper runs Overlap Gate checks (Ch. 5; Ch. 12 T14) on intersections and builds a coverage graph of reverse-validated cells.
5. **Search-layer report.** If the union of reverse-validated cells covers the target domain and no barrier remains, emit a **Search-layer Coverage Report** with full logs and certificates. This report is not a Core theorem.

Remark 11.42 (Non-claims). A successful Program B2 run does not prove analytic regularity or number-theoretic identities. It is a reproducible Search-layer artifact asserting that, *under the declared gates and tests*, the pipeline found no surviving barriers on the explored domain.

16.4. Program B3: Counterexample hunt at the Search Layer [Spec]

Specification 11.43 (Program B3: robust barrier discovery). **Goal:** search for stable Peak-regime regions (Type IV) that persist across admissible repairs and lifts.

Procedure:

1. **Maximization / targeting.** Hunters target points where Φ_τ is large, prioritizing cells that repeatedly fail UCC budgets.
2. **Repair attempts.** Apply Restart/refinement (Ch. 12 T11) and admissible deletion/inclusion updates (Ch. 11/12 T2) to rule out discretization artifacts.
3. **Lifting attempts.** Invoke the Lifter under the Lifting Penalty (Ch. 14), recording whether peaks disappear under admissible lifts.
4. **Candidate output.** If a Peak persists across admissible refinements and lifts and continues to show $(\mu, u) \text{ eq}(0, 0)$ at declared scales, output a **Certified Counterexample Candidate (Search-layer)** with complete artifacts: `bars/spec/aux/ext/phi.json` and the manifest `run.yaml`.

Such candidates require independent mathematical analysis outside the automated pipeline.

16.5. Summary and guard-rails

Remark 11.44 (Summary). Bridge Programs B1–B3 formalize the Search-layer handling of reverse uses:

- **B1 (A-plan)** allows reverse propagation only in the pre-audited definable_or_saturation scope and only when the mandatory named test $T\text{-ExtZero} \Rightarrow \text{PHZero}$ passes (Ch. 12, §12.2a).
- **Spectral gaps** are optional auxiliary safety checks; they do not expand scope.
- **B2/B3** define Search-layer coverage vs. barrier discovery workflows using Hunter/Mapper/Lifter.

All results remain within Part II and do not modify Part I theorems. No claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made.

Chapter 17: The AK-HDPST AI Platform

17.0. Overview: from theory to execution

The preceding chapters established (i) the mathematical Core (Part I) and (ii) the Search Layer protocols (Part II), including the UCC ledger, test suite, Bridge Programs (A-plan), and agent behaviors. This chapter specifies the *execution environment*—the AK-HDPST AI Platform.

In this framework, a “proof artifact” is not a static text but a *reproducible computational process*: a manifest plus verifiable artifacts that can be re-executed by an independent auditor. To preserve rigor, we enforce a strict separation of roles:

- **AI Agents (Proposers):** untrusted agents (Hunters / Mappers / Lifters) proposing points, moves, restarts, and lifts.
- **AK Core (Verifier):** a trusted deterministic kernel implementing Part I, the UCC logic, and the formal test suite (Ch. 12), auditing every proposal and issuing accept/reject verdicts.

Standing convention. All persistence-layer quantities used for acceptance are computed *after* \mathbf{T}_τ (equivalently on $C_\tau F$), and all comparisons follow the unique order **T–PFBC–AfterCollapse** (Ch. 12).

17.1. run.yaml as a formal proof object (manifest + artifacts)

Definition 11.45 (Proof Object). A *Proof Object* in AK-HDPST is a tuple

$$\mathcal{P} = (\text{run.yaml}, \mathcal{A}, \mathcal{H}),$$

where `run.yaml` is a manifest, \mathcal{A} is the set of referenced artifacts (barcodes, spectra, Ext-data, comparison maps, logs), and \mathcal{H} is the set of cryptographic hashes binding the manifest to the artifacts and to the Core implementation/version used in the run.

Specification 11.46 (Manifest conformance and mandatory fields). A `run.yaml` is *conformant* if it satisfies the schema requirements of Ch. 12 (Decl. 11.1) and records at least:

- coefficient field and backend policy;
- τ -sweep specification and window policies;
- quantale block `quantale:{name, op, unit, order, mode}`;
- layered deltas `layered_delta:{deltaGal, deltaTr, deltaFun}` when applicable;
- definable windows `definable:{structure, window_formulae}`;
- Iwasawa block and AWFS/2-cell block when enabled;
- spectral policy (order: ascending, norm declaration) and bounds;
- Restart/Summability evidence and Overlap Gate fields;
- A/B commutativity policy and tolerance;
- explicit (μ, u) totals, `phi_iso_tail`, and pass criteria (per-degree vs. aggregated).

Specification 11.47 (Validity of a Proof Object (deterministic verification)). A Proof Object \mathcal{P} is *valid* if and only if the following hold under deterministic re-execution by the AK Core (same Core hash/version as recorded in \mathcal{H}):

1. **MECE coverage (finite or countable).** The referenced Terrain Cells form a declared MECE cover of the target domain, with event accounting and coverage checks (Ch. 12, T9; countable via T-Countable-Cover).
2. **Cell-level acceptance (UCC).** Every cell marked passed: true includes an AK Core certificate that verifies UCC acceptance (gap check and ledger bound) and the tower diagnostics policy (including Type IV handling via (μ, u)) at the declared τ .
3. **Gluing and overlap.** All claimed adjacencies/merges are supported by Overlap Gate logs and artifacts (Ch. 12, T14), including A/B soft-commuting or fallback with δ^{alg} charging (Ch. 12, T10).
4. **Restart/Summability.** The global certificate includes Restart constants and Summability evidence (Ch. 12, T11 and T-Delta-Sum-Converges), preventing hidden accumulation of defects.
5. **Reverse certificates (Bridge Program A-plan).** Any cell labeled reverse_certified: true must satisfy:
 - scope definable_or_saturation (Ch. 16, Program B1 A-plan),
 - the mandatory named test T-ExtZero->PHZero is applicable and passes (Ch. 12, §12.2a; alias T7/T17),
 - CCC fields and artifacts are present (tower stability, potential bound, logs).

Optional spectral-gap checks may be recorded as spectral_aux_ok, but *cannot* make an inapplicable reverse certificate valid.

Verification reduces to re-running run.yaml against the AK Core; acceptance is deterministic given the recorded hashes and policies.

Remark 11.48 (Artifact minimal set). A minimal auditable bundle referenced by run.yaml includes bars.json, spec.json, ext.json, phi.json, and the stepwise logs (e.g. hunter_log.jsonl, mapper_log.jsonl), optionally an .h5 container for large runs (Ch. 12, §12.3).

17.2. Reproducibility: Hunter Action Log schema (with gradient norm discipline)

Definition 11.49 (Hunter Action Log). A *Hunter Action Log* is a sequential record $\mathcal{L} = (S_0, A_0, S_1, A_1, \dots)$ where S_k is the state and A_k is the action at step k , sufficient to replay the exploration under the same manifest and Core.

Specification 11.50 (Required fields for replayability). A valid Hunter log must contain at least:

1. **Initialization.** RNG seeds; initial point x_0 ; initial Terrain Cell W_0 ; hyperparameters $(\alpha, \text{step caps}, \lambda_{\text{sing}}, \text{restart thresholds})$.
2. **Regime classification data.** For each step: $\Phi_\tau(x_k)$, the regime label (Plain/Ridge/Peak), the declared τ , and whether acceptance checks were invoked.
3. **Approximate gradient specification (mandatory when used).** If Navigation Mode returns an *approximate gradient direction*, the log must record the following as [Spec]:

- **Estimator type:** finite differences / random perturbation / surrogate model;
 - **Stencil / perturbation details:** neighbor set, step radius, distribution, sample count;
 - **Norm choice:** which norm on \mathcal{M} is used for Lipschitz control and step normalization;
 - **Estimator error charging:** the gradient estimation error (and sampling residuals) is charged to $\delta_{\text{meas}} \oplus \delta_{\text{disc}}$ (and logged distinctly from δ^{alg});
 - **Acceptance rule:** whether the step is accepted by descent check and/or Core veto.
4. **Restart trace.** When restart/refinement occurs: the trigger condition, the refined cells W' , updated local thresholds (if any), and the linkage to Restart/Summability evidence.
 5. **Lifting trace.** For each committed lift: axis type, local one-sided derivative / finite difference along the axis, charged Lifting Penalty δ^{lift} , lifted coordinates, and the new Hunter spawn record.
 6. **Cross-check hooks.** References to artifacts produced/consumed at each step: bars/spec/ext/phi.json pointers and their hashes.

Replayability requirement. Given the same seeds, the same run.yaml, and the same Core hash/version, a third party must be able to replay \mathcal{L} and recover the same set of validated cells and certificates, up to declared numerical tolerances.

17.3. Architecture: proposer–verifier separation (white-box audit)

The AK-HDPST AI Platform adheres to a strict Proposer–Verifier model:

- **Proposer.** AI agents propose candidate moves: points x , schedules of τ -sweeps, restart/refinement steps, and candidate lifts.
- **Verifier (AK Core).** The Core computes Φ , updates the δ -ledger, evaluates tower diagnostics (μ, u) , runs formal tests (Ch. 12), checks gates (B-Gate⁺, Overlap Gate, CCC, saturation/definable scopes), and returns accept/reject for each proposed certificate.
- **Proof store.** Accepted cell certificates, overlap edges, and global coverage evidence are written to the proof store and referenced by hashes in run.yaml.

Remark 11.51 (White-box principles). The platform enforces:

1. **AI visibility only.** Agents may observe diagnostics and potential values but cannot modify Core thresholds, comparison orders, or acceptance rules.
2. **Core sovereignty.** If $\Sigma\delta \not\leq \text{gap}_\tau$, if Type IV policies fail, or if mandatory tests do not pass, the Core vetoes the certificate regardless of any agent confidence.
3. **No black-box acceptance.** Claims are accepted only as a consequence of deterministic Core verification of the manifest and artifacts.

17.4. Platform outputs: Map of Validity (as a verifiable object)

Definition 11.52 (Map of Validity). A *Map of Validity* is the union of: (i) a MECE cell decomposition of the target domain, (ii) cell labels (valid, undecided, barrier), (iii) overlap edges and connected components produced by the Mapper, and (iv) the complete set of certificates and logs referenced in run.yaml.

Remark 11.53 (Interpretation and non-claims). The Map of Validity is a *Search-layer* object: it is auditable, replayable, and bounded by the implementable-range rules and tests. It does not, by itself, assert new Core theorems (e.g. analytic regularity or number-theoretic identities). Any global mathematical conclusion requires an explicit statement of scope, gates, and tests used to construct the map.

17.5. Final conclusion

AK-HDPST v17.0 completes the transition from a static theoretical framework to a *computational proof engine* in the implementable range. By combining:

1. the rigorous **Collapse Core** (Part I),
2. the disciplined **Search Layer** with formal tests and A-plan Bridge Programs (Part II),
3. and the auditable **AI Platform** (Chapter 17),

the theory supports large-scale exploration of parameter spaces via a reproducible, verifier-certified pipeline.

In this setting, the central deliverable is a **Map of Validity**: a high-dimensional, replayable certificate bundle proposed by agents and certified by the AK Core under UCC, the test suite, and strict guard-rails.

““tex

Notation and Conventions (reinforced v17.0)

Base field and ambient categories. Fix a coefficient field k (Appendices N/O may instead use a field Λ ; when used, replace k by Λ everywhere). Let Vect_k be the abelian category of finite-dimensional k -vector spaces and write $[\mathbb{R}, \text{Vect}_k]$ for functors $(\mathbb{R}, \leq) \rightarrow \text{Vect}_k$.

Constructible persistence and standing identification. We write

$$\text{Pers}_k^{\text{cons}} \subset [\mathbb{R}, \text{Vect}_k]$$

for the full subcategory of *constructible* persistence modules (pointwise finite-dimensional with locally finite critical set on bounded windows). Throughout the paper we *identify* the “finite-type” category with this constructible subcategory and use the symbol $\text{Pers}_k^{\text{cons}}$ uniformly.

Badges and layer discipline (Core vs. Search). Statements are tagged by their layer: $[\text{Prop}]/[\text{Thm}]$ are reserved for Part I (Core) results; $[\text{Declaration}]/[\text{Spec}]$ record operational commitments (tests, manifests, logging, search heuristics); $[\text{Conjecture}]$ denotes forward-looking, non-audited claims. Reverse inferences (e.g. $\text{Ext} \Rightarrow \text{PH}$) are *never* asserted as Core theorems; if used, they appear only as Search-layer specifications with explicit certificates and ledger charges.

Filtered objects and persistence. $\text{FiltCh}(k)$ denotes filtered chain complexes of finite-dimensional k -spaces; filtered quasi-isomorphism is abbreviated f.q.i. For $i \in \mathbb{Z}$ the degree- i persistence functor is

$$\mathbf{P}_i : \text{FiltCh}(k) \longrightarrow \text{Pers}_k^{\text{cons}}, \quad \mathbf{P}_i(F)(t) = H = H(F^t).$$

Realizations from other formalisms into $\text{FiltCh}(k)$ are denoted $\mathcal{R}(-)$ (and $\mathcal{F}(-)$, etc. as appropriate).

Quantale enrichment, δ -ledger, and definable windows (UCC layer). Unless stated otherwise, the measurement layer is enriched over a fixed *commutative quantale* V (write $(V, \oplus, \leq, 0)$). Distances are Lawvere V -valued, and all aggregates (budgets, residuals, tolerances) are taken using the single operation \oplus (“quantale-sum”). The δ -ledger is recorded in (possibly product) quantales and is decomposed as

$$\delta = \delta_{\text{alg}} \oplus \delta_{\text{disc}} \oplus \delta_{\text{meas}}, \quad \text{with optional layered bookkeeping } (\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}}) \text{ when enabled.}$$

Window sets are *right-open* and MECE along a τ -sweep, and—when required—*Denef–Pas definable*; definability guarantees finite event sets and finite Čech depth on bounded windows (Appendix Q; cf. Appendices H/J for the real o-minimal surrogate $\mathbb{R}_{\text{an,exp}}$). The triple $(V, \text{definable windows, AWFS/2-cells})$ is referred to as the *Unified Collapse Contract (UCC)*.

Reflection/truncation vs. window clipping. For $\tau \geq 0$, let $E_\tau \subset \text{Pers}_k^{\text{cons}}$ be the Serre subcategory generated by bars of length $\leq \tau$. The *reflector* (bar-deletion truncation)

$$\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \longrightarrow (E_\tau)^\perp$$

is exact, idempotent, and left adjoint to the inclusion $(E_\tau)^\perp \hookrightarrow \text{Pers}_k^{\text{cons}}$ (Appendix A, Theorem .64); it is 1-Lipschitz for d_{int} and, more generally, V -1-Lipschitz in the V -enriched metric (Appendix A, Proposition .70). On filtered complexes we use a colapser C_τ with a natural (up to f.q.i.) identification

$$\mathbf{P}_i(C_\tau F) \cong \mathbf{T}_\tau(\mathbf{P}_i F) \quad (\text{natural in } F, i).$$

For a right-open window $W = [u, u')$, the *window clip* is denoted \mathbf{W}_W (a.k.a. W_{clip} ; Definition 5.4): it restricts to W and extends by zero; it is exact and 1-Lipschitz for d_{int} (Appendix A and Chapter 6). *Warning.* \mathbf{T}_τ (delete bars of length $\leq \tau$) and \mathbf{W}_W (clip to a window) play distinct roles and must not be conflated.

Optional safe low-pass (after-collapse only). A safe low-pass operator LP_τ may be applied *after* \mathbf{T}_τ and \mathbf{W}_W , but *only* to spectral auxiliaries derived from $L(C_\tau F|_W)$ (Chapter 6.0bis, Definition 5.5). It never modifies the persistence-layer objects $\mathbf{T}_\tau \mathbf{P}_i(F|_W)$ used for gates and certification; non-compliant filters are not adopted for certification.

AWFS viewpoint and 2-cells. When enabled, we use an algebraic weak factorization system on $\text{Ho}(\text{FiltCh}(k))$,

$$\text{Id} \Rightarrow L \dashv R \Rightarrow \text{Id}, \quad R = C_\tau,$$

to organize preprocessing/realization. Any resulting 2-cell defects are accounted for as algorithmic ledger entries δ_{alg} (Chapter 5; Appendices K/L) and logged in the run manifest (Appendix G).

Interleaving metric and shifts. On $\text{Pers}_k^{\text{cons}}$ the interleaving metric d_{int} equals the bottleneck distance in the constructible 1D setting. The time shift is $(S^\varepsilon M)(t) := M(t + \varepsilon)$; shifts commute canonically with \mathbf{T}_τ , hence \mathbf{T}_τ is 1-Lipschitz (and V -1-Lipschitz when the measurement layer is V -enriched).

Unique comparison order (after-collapse policy). All comparisons (local, overlap, global) follow the unique order

$$\boxed{\text{for each } t \Rightarrow \mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare in } \text{Pers}_k^{\text{cons}}}.$$

Equivalently, comparisons are performed *after applying* \mathbf{T}_τ (equivalently on $C_\tau F$) and *never* on pre-collapse objects. This order is mandatory for audits, spectral alignment, and overlap gluing (Chapter 11, §11.0 bis; Chapter 5).

Barcodes, events, and endpoint convention. Barcodes use half-open intervals $I = [b, d)$ with $d \in \mathbb{R} \cup \{\infty\}$ and multiplicity $m(I) \in \mathbb{Z}_{\geq 1}$. Any consistent open/closed choice yields the same clipped lengths and event sets. For $\tau \geq 0$ the clipped length is

$$\ell_{[0, \tau]}(I) := \max\{0, \min\{d, \tau\} - \max\{b, 0\}\}.$$

Given $\tau_0 > 0$, the finite event set in degree i is

$$\text{Ev}_i(F; \tau_0) = \{0, \tau_0\} \cup (\{b \in [0, \tau_0]\} \cap \text{births}) \cup (\{d \in [0, \tau_0]\} \cap \text{deaths}),$$

with endpoint conventions and infinite bars as in Appendix A, Remark .63.

Betti curves and Betti integral. $\beta_i(F; t) := \dim_k H_i(F^t)$ is càdlàg and piecewise constant on bounded windows. The (clipped) Betti integral is

$$\text{PE}_i^{\leq \tau}(F) = \int_0^\tau \beta_i(F; t) dt = \sum_{I \in \mathcal{B}_i(F)} m(I) \ell_{[0, \tau]}(I) \quad (\text{Appendix H}).$$

Length spectrum operator. For $M \in \text{Pers}_k^{\text{cons}}$ with barcode $M \simeq \bigoplus_j I[b_j, d_j)$ and a right-open window $W = [u, u')$, the *length spectrum* operator $\Lambda_{\text{len}}(M; W)$ (Chapter 2) is diagonal on a bar-basis with eigenvalues $\ell_W(I[b_j, d_j))$. Its multiset of eigenvalues equals the multiset of clipped bar lengths and is invariant under isomorphisms $M \simeq M'$. In particular, for $\mathbf{T}_\tau \mathbf{P}_i(F)$ and $W = [0, \tau]$,

$$\text{PE}_i^{\leq \tau}(F) = \|\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(F); [0, \tau])\|_{L^1},$$

hence the total collapse energy is an isomorphism invariant of the truncated persistence (Chapter 11, §11.0+).

Spectral policy, ordering, and matrix norms (after-collapse only). Spectral indicators are computed on $L(C_\tau F)$ (normalized combinatorial Hodge Laplacian), never on pre-collapse objects. Positive eigenvalues are reported in *ascending* order. The matrix norm used for tolerances is declared as $\|\cdot\|_{\text{op}}$ (operator) or $\|\cdot\|_{\text{fro}}$ (Frobenius) and recorded in the run manifest (Appendix G). Allowed uses of spectrum are limited to: (i) deletion-type non-increase and (ii) ε -continuation stability under d_{int} (Appendix E); no analytic/geometric conclusions are asserted beyond these audited regimes.

Towers and diagnostics; $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)/(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$ **and Type IV.** A *tower* is a directed system $F = (F_n)_{n \in I}$ with colimit F_∞ . For $i \in \mathbb{Z}$, $\tau \geq 0$, the comparison map is

$$\phi_{i, \tau}(F) : \varinjlim_n \mathbf{T}_\tau(\mathbf{P}_i(F_n)) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(F_\infty)).$$

Set $\mu_{i, \tau}(F) := \text{gdim ker } \phi_{i, \tau}(F)$ and $u_{i, \tau}(F) := \text{gdim coker } \phi_{i, \tau}(F)$; the totals are

$$\mu_{\text{Collapse}}(F) := \sum_i \mu_{i, \tau}(F), \quad u_{\text{Collapse}}(F) := \sum_i u_{i, \tau}(F).$$

We write $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ for $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$ for $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$. The regime $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$ is the *Type IV (invisible defect)* diagnostic of v17.0. Cofinal restriction leaves $((\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)/(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0))$ unchanged; finite direct sums add; composition is subadditive and the rules extend to V -distances (Appendix J). If $\phi_{i, \tau}$ is an isomorphism for all relevant i , then $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$. The diagnostic μ_{Collapse} (and u_{Collapse}) is a persistence-layer obstruction and is *distinct* from the classical Iwasawa μ -invariant.

Gate cascade (audited evaluation order; no illicit reverse theorems). The default gate cascade is *operated* as

$$E_1=0 \implies (\mu, u)=(0,0) \implies \text{Ext}^1=0 \implies \text{PH}_1=0,$$

as a sequent calculus with cut elimination (Chapter 6.3bis; Chapter 1/3/5): success at a later stage never overturns failure at an earlier stage. Logical implications are only claimed where explicitly proved or certified; in particular, no global equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is asserted.

Overlap Gate (local→global gluing). Given a windowed cover $\{(X_\alpha, W_\alpha)\}$ by right-open *definable* windows, the *Overlap Gate* (Chapter 1; Chapter 5) requires, after truncation by \mathbf{T}_τ ,

- local post-collapse equivalence on overlaps up to the recorded δ -budget (quantitative commutation);
- Čech– Ext^1 acyclicity in degree 1 (finite Čech depth holds by definability);
- stability-band condition $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and near- τ non-accumulation.

When all overlaps pass, local truncated objects glue (uniquely up to isomorphism) in $\text{Pers}_k^{\text{cons}}$. Run manifests record these under `overlap_checks` (Appendix G): `local_equiv`, `cech_ext1_ok`, `stability_band_ok`.

A/B soft-commuting; Mirror/Transfer; pipeline budget; B-Gate⁺. For exact reflectors T_A, T_B in $\text{Pers}_k^{\text{cons}}$, the commutation defect is

$$\Delta_{\text{comm}}(M; A, B) = d_{\text{int}}(T_A T_B M, T_B T_A M).$$

Given tolerance η , if $\Delta_{\text{comm}} \leq \eta$ we accept *soft-commuting*; else fix an order (e.g. $T_B \circ T_A$) and *record* Δ_{comm} as δ_{alg} (Appendices K/L). For Mirror/Transfer functors Mirror, assume a natural 2-cell $\text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$ with a uniform bound $\delta(i, \tau) \geq 0$ in d_{int} , additive along pipelines and *non-increasing* under 1-Lipschitz post-processing (Appendix L). On a window W and degree i , the pipeline budget aggregates (in the quantale)

$$\Sigma\delta(i, \tau) = \bigoplus_{\text{Mirror-Collapse}} \delta(i, \tau) \oplus \bigoplus_{\text{A/B fails}} \Delta_{\text{comm}} \oplus \bigoplus_{\text{audits}} (\delta_{\text{disc}} \oplus \delta_{\text{meas}}).$$

The *safety margin* $\text{gap}_\tau > 0$ is configured per window and degree; B-Gate⁺ requires $\text{gap}_\tau > \Sigma\delta(i, \tau)$. Across windows, Restart and Summability hold if there exists $\kappa \in (0, 1]$ with

$$\text{gap}_{\tau_{k+1}} \geq \kappa(\text{gap}_{\tau_k} - \Sigma\delta_k(i)), \quad \bigoplus_k \Sigma\delta_k(i) < \infty$$

(Appendix J), with the second condition interpreted in the chosen quantale.

Arithmetic alignment conventions (Iwasawa Gate). When applicable, the Iwasawa diagnostic μ_{Iwasawa} is compared with the collapse diagnostic $\mu_{\text{Collapse}} := \mu_{\text{Collapse}}$ in a three-state regime (lower bound / match / drift-corrected) used by the Gate cascade to suppress Type-IV drift (Chapter 7; Appendix R). Finite kernel/-cokernel effects are absorbed into δ_{alg} .

Categorical check and window-local trigger. With $Q := \{k[0]\}$, we test $\text{Ext}^1(\mathcal{R}(C_\tau F), Q) = 0$ *after truncation*. The one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ is used only under (B1)–(B3) (field coefficients; constructible range; t -exact realization of amplitude ≤ 1). On a definable right-open window W of finite Čech depth we adopt the *window-local trigger* via the first energy page E_1 :

$$E_1(W) = 0 \iff \text{PH}_1(C_\tau F|_W) = 0 \iff \text{Ext}^1(\mathcal{R}(C_\tau F)|_W, k) = 0,$$

while globally only sanctioned one-way implications (and certified pasting via Restart/Summability) are used.

Reproducibility (manifest) and mandatory fields. The run manifest `run.yaml` (v17) is treated as a proof object (Chapter 17; Appendix G) and must include, at minimum: `version`, `suite_version`, `seed`; UCC hooks `quantale`, `definable`, `awfs_2cell`, `layered_delta`; windows (right-open, MECE, and coverage), `coverage_check`; operations (including A/B commuting policy and fallbacks), budget (including `gap_tau` and \oplus -totals), gate and overlap_checks; persistence (PH/Ext/(μ_{Collapse} , u_{Collapse}) and tail isomorphism flags); `spectral_policy` with mandatory order: "ascending" and norm: "op"|"fro" plus any declared bounds/tolerances; and, when enabled, `iwasawa` and other cross-domain control blocks. All persistence-layer quantities are computed after applying \mathbf{T}_τ (equivalently on $C_\tau F$); spectral auxiliaries follow the fixed normalization policy; any residual slack is charged to the ledger and logged.

Global guard-rails and non-claims. All statements are confined to the persistence/spectral/categorical layers in the implementable range. No number-theoretic identity, analytic regularity theorem, or group trivialization is asserted. No claim of $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made; only the one-way implication $\text{PH}_1 \Rightarrow \text{Ext}^1$ (under (B1)–(B3)) is used. The collapse obstruction μ_{Collapse} is a persistence-level diagnostic and is distinct from the classical Iwasawa μ .

Abbreviations. f.q.i. = filtered quasi-isomorphism; $\text{c\grave{a}dl\grave{a}g}$ = right-continuous with left limits; UCC = Unified Collapse Contract; AWFS = algebraic weak factorization system; DP = Denef–Pas definable; LP_τ = safe low-pass; “window” = right-open interval $W = [u, u')$ (often $W = [0, \tau]$). “

Appendix A. Constructible Persistence: Abelianity, Serre Localization, and the V-Nucleus View (reinforced v17.0)

Throughout this appendix, fix a field k . Write Pers_k for the category of right-continuous persistence modules $M : (\mathbb{R}, \leq) \rightarrow \text{Vect}_k$ with structure maps $M(t \leq t')$. We denote by $\text{Pers}_k^{\text{ft}} \subset \text{Pers}_k$ the *constructible* (finite-type) subcategory used in the main text.

Global conventions (v17.0 alignment). (i) All Ext-tests are taken against $Q = k[0]$ (the unit interval module supported at a point), unless explicitly stated otherwise.

(ii) Windowed energies use an exponent $\alpha > 0$ (default $\alpha = 1$).

(iii) References to appendices use the tilde style (e.g. Appendix D); failure types use the dash style Type I–II, Type III, Type IV.

(iv) Notational disambiguation: the reflector (Serre localization) functor is denoted \mathbf{T}_τ ; truncations/clippings on domains are denoted \mathbf{Tr}_τ or $\text{clip}_{[a,b]}$. This resolves the potential collision sometimes found in informal notes where \mathbf{T}_τ was used for truncation.

(v) When window partitions are used (MECE; §A.6), *half-open*, *right-open* (i.e. left-closed/right-open) is the standing endpoint convention for domain windows and spectral bins; coverage checks are mandatory and any discretization slack is recorded (Appendix G).

(vi) All results in §§A.2–A.5 are stated and used in the one-parameter (1D), field-coefficient, right-continuous, constructible setting.

(vii) *δ -ledger interface.* Quantitative non-commutation and implementation defects are externalized as

$$\delta = \delta_{\text{alg}} \oplus \delta_{\text{disc}} \oplus \delta_{\text{meas}} \in V,$$

summed in a fixed commutative quantale V (see §A.0). Windowed pipelines aggregate δ ’s using \oplus . When the layered ledger is enabled, entries may additionally be reported in boxed form $(\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}})$ and then aggregated in the product quantale (Appendix S; schema: Appendix G).

Remark .54 (Scope / after-collapse alignment). Appendix A asserts equalities and exactness statements purely at the persistence layer $\text{Pers}_k^{\text{ft}}$. All *operational* quantitative comparisons (distances, Lipschitz constants, monotonicity, budgets) used in the paper are evaluated in the mandated order $\mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{Pers}_k^{\text{ft}}$ on fixed right-open windows (after-collapse policy). Equivalently, whenever the main text views these objects via a filtered model F , all such comparisons are made *after applying* \mathbf{T}_τ (equivalently on $C_\tau F$) and then returned to $\text{Pers}_k^{\text{ft}}$; no filtered-level equalities are asserted in this appendix.

A.0. V -nucleus viewpoint and the δ -interface

Let V be a commutative quantale $(V, \oplus, \leq, 0)$, and endow $\text{Pers}_k^{\text{ft}}$ with a Lawvere V -metric d_V induced by the interleaving (bottleneck) distance. (The canonical choice is $V = [0, \infty]$ with $\oplus = +$ and the usual order \leq .)

Definition .55 (V -nucleus (Lawvere setting)). A V -nucleus on a V -metric category (C, d_V) is an endofunctor $N : C \rightarrow C$ together with a natural transformation $\eta : \text{Id} \Rightarrow N$ such that:

- (Idempotence) $N \circ N \simeq N$.
- (Non-expansiveness / V -1-Lipschitz) $d_V(NX, NY) \leq d_V(X, Y)$ for all X, Y .
- (Exactness when an abelian structure is present) N is exact on the underlying abelian category (equivalently, preserves finite limits and finite colimits).

Proposition .56 (\mathbf{T}_τ is a V -nucleus). *In the constructible 1D range, the Serre reflector $\mathbf{T}_\tau : \text{Pers}_k^{\text{ft}} \rightarrow \text{Pers}_{k, \tau\text{-loc}}^{\text{ft}}$ (Theorem .64) is a V -nucleus in the sense of Definition .55: it is idempotent (Cor. .65), 1-Lipschitz for interleavings (Prop. .70), and exact (Theorem .64).*

Remark .57 (δ -commutation schema). At the categorical level, functors that preserve the τ -ephemeral Serre class commute strictly with \mathbf{T}_τ . Operationally, we record any implementation-level discrepancy by a non-negative $\delta \in V$, decomposed as in (vii). For clipping on a right-open window $W = [u, v)$ we define a *clipping defect* $\delta_{\text{clip}}(W, \tau) \in V$ with the contract

$$d_V(\mathbf{T}_\tau \circ \text{clip}_W(M), \text{clip}_W \circ \mathbf{T}_\tau(M)) \leq \delta_{\text{clip}}(W, \tau),$$

which must be logged (Appendix G) and is *subadditive* over pipelines and windows (with respect to \oplus). By default, δ_{clip} is charged to $\delta_{\text{disc}} \oplus \delta_{\text{meas}}$ unless an algorithmic 2-cell origin is explicitly declared (Appendices K/L). In the ideal (exact) model, $\delta_{\text{clip}}(W, \tau) = 0$ (Lemma .74); any nonzero value reflects discretization/rounding policies and is accounted for in the δ -ledger.

A.1. Constructible objects

Definition .58 (Constructible / finite-type). A persistence module $M \in \text{Pers}_k$ is *constructible* (finite-type) if on every bounded interval $[a, b] \subset \mathbb{R}$ it has a *finite critical set*: there exist $a = t_0 < t_1 < \dots < t_N = b$ such that each structure map $M(t \leq t')$ is an isomorphism whenever $t, t' \in (t_j, t_{j+1})$ for some j . Equivalently, M is pointwise finite-dimensional and admits a barcode decomposition as a *locally finite direct sum of interval modules*, i.e. only finitely many intervals intersect any bounded window. We write $\text{Pers}_k^{\text{ft}}$ for the full subcategory of such modules.

Remark .59. In the 1D, field-coefficient, right-continuous setting, the equivalence above is standard (barcode decomposition). All constructions below (kernels, cokernels, torsion, Serre localization, clipping) preserve constructibility and are controlled by finitely many events on bounded windows; see the references at the end of this appendix.

A.2. Abelianity

Proposition .60. $\text{Pers}_k^{\text{ft}}$ is an abelian category. Moreover, for a morphism $f : M \rightarrow N$ in $\text{Pers}_k^{\text{ft}}$, kernels and cokernels are computed pointwise in Vect_k and remain constructible.

Proof. Evaluation at each $t \in \mathbb{R}$ is exact in Vect_k , hence pointwise kernels and cokernels define functorial sub/quotient persistence modules. Constructibility is preserved: on any bounded window one refines the break sets of M, N to a finite set controlling $\text{Ker } f$ and $\text{Coker } f$. Exactness axioms follow objectwise; hence $\text{Pers}_k^{\text{ft}}$ is abelian with pointwise exactness. \square

A.3. The τ -ephemeral Serre subcategory

Fix $\tau > 0$. Let $I[a, b)$ denote the interval module supported on $[a, b)$ (half-open, right-open convention).

Definition .61 (τ -ephemeral subcategory). Let $E_\tau \subset \text{Pers}_k^{\text{ft}}$ be the smallest full subcategory containing all interval modules $I[a, b)$ with length $b - a \leq \tau$ and closed under subobjects, quotients, and extensions. We call E_τ the τ -ephemeral (or τ -torsion) subcategory.

Lemma .62. E_τ is a Serre subcategory of $\text{Pers}_k^{\text{ft}}$, and it is hereditary as a torsion class.

Remark .63 (Endpoint conventions). All statements in this appendix are insensitive to the choice of open/closed endpoints on interval modules. We fix $[a, b)$ for definiteness; changing endpoint conventions does not affect lengths, barcode decompositions, interleaving/bottleneck distances, or any categorical constructions below.

A.3.1. The torsion pair and maximal τ -ephemeral subobject

Define the τ -local (orthogonal) subcategory

$$\text{Pers}_{k, \tau\text{-loc}}^{\text{ft}} := \{ X \in \text{Pers}_k^{\text{ft}} \mid \text{Hom}(E, X) = 0 = \text{Ext}^1(E, X) \text{ for all } E \in E_\tau \}.$$

Then $(E_\tau, \text{Pers}_{k, \tau\text{-loc}}^{\text{ft}})$ is a torsion pair: for each M there is a functorial short exact sequence

$$0 \longrightarrow t_\tau(M) \longrightarrow M \longrightarrow f_\tau(M) \longrightarrow 0$$

with $t_\tau(M) \in E_\tau$ and $f_\tau(M) \in \text{Pers}_{k, \tau\text{-loc}}^{\text{ft}}$.

A.4. The reflector $T_\tau \dashv \iota_\tau$, exactness, and the V -nucleus corollary

Let $\iota_\tau : \text{Pers}_{k, \tau\text{-loc}}^{\text{ft}} \hookrightarrow \text{Pers}_k^{\text{ft}}$ be the inclusion.

Theorem .64 (Exact reflective localization). *The Serre quotient functor*

$$\pi_\tau : \text{Pers}_k^{\text{ft}} \longrightarrow \text{Pers}_k^{\text{ft}}/E_\tau$$

is exact. In the 1D constructible setting there is a canonical exact equivalence of abelian categories

$$\text{Pers}_k^{\text{ft}}/E_\tau \simeq \text{Pers}_{k, \tau\text{-loc}}^{\text{ft}}.$$

Composing π_τ with this equivalence yields a functor

$$T_\tau : \text{Pers}_k^{\text{ft}} \longrightarrow \text{Pers}_{k, \tau\text{-loc}}^{\text{ft}}$$

which is left adjoint to ι_τ and is exact.

Corollary .65 (Idempotence, conservativity, V -nucleus). $\mathbf{T}_\tau \circ \mathbf{T}_\tau \cong \mathbf{T}_\tau$ and $\mathbf{T}_\tau|_{\text{Pers}_{k,\tau\text{-loc}}^{\text{ft}}} \cong \text{Id}$. Together with Proposition .70, \mathbf{T}_τ is a V -nucleus (Definition .55) as recorded in Proposition .56.

Proposition .66 (Behavior on barcodes). Let $M \simeq \bigoplus_j I[a_j, b_j]$. Then

$$\mathbf{T}_\tau M \simeq \bigoplus_{b_j - a_j > \tau} I[a_j, b_j], \quad t_\tau(M) \simeq \bigoplus_{b_j - a_j \leq \tau} I[a_j, b_j].$$

Remark .67 (Filtered colimits: functor-category computation and return to constructible). Filtered colimits are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$, and \mathbf{T}_τ commutes with those colimits there (as a left adjoint). A filtered colimit of constructible modules may exit $\text{Pers}_k^{\text{ft}}$. In applications we either: (i) restrict to towers that remain constructible degreewise; or (ii) compute in $[\mathbb{R}, \text{Vect}_k]$, apply \mathbf{T}_τ , and *verify* return to $\text{Pers}_k^{\text{ft}}$ on each declared window (coverage + finiteness checks logged in run.yaml; Appendix G).

A.5. Shift-commutation, monotonicity in τ , and 1-Lipschitz continuity

For $\varepsilon \geq 0$, let S^ε be the shift $(S^\varepsilon M)(t) := M(t + \varepsilon)$.

Lemma .68 (Shift commutation). For all $\varepsilon \geq 0$, $\mathbf{T}_\tau \circ S^\varepsilon \cong S^\varepsilon \circ \mathbf{T}_\tau$.

Lemma .69 (Monotonicity in τ). If $0 < \tau \leq \tau'$, there is a natural epimorphism $\mathbf{T}_{\tau'} M \rightarrow \mathbf{T}_\tau M$, functorial in M .

Proposition .70 (Non-expansiveness (interleaving/bottleneck)). \mathbf{T}_τ is 1-Lipschitz for the interleaving (equivalently, bottleneck) distance on $\text{Pers}_k^{\text{ft}}$. Equivalently, \mathbf{T}_τ is V -1-Lipschitz for the induced Lawvere V -metric d_V in the canonical $V = [0, \infty]$ regime.

A.6. Windowing (MECE), coverage checks, and τ -adaptation

Definition .71 (MECE domain windowing and coverage). A *domain windowing* is a finite or countable collection of half-open, right-open intervals $\{[u_k, u_{k+1})\}_{k \in K}$ such that:

- (Disjointness) $[u_k, u_{k+1}) \cap [u_\ell, u_{\ell+1}) = \emptyset$ for $k \neq \ell$.
- (Contiguity) $u_{k+1} = u_k + \text{len}_k$ with $\text{len}_k > 0$.
- (Coverage) $\bigsqcup_{k \in K} [u_k, u_{k+1}) = [u_0, U)$ for some finite $U > u_0$.

Coverage checks require

$$\sum_{k \in K} (u_{k+1} - u_k) = U - u_0, \quad \# \text{Events}([u_0, U)) = \sum_{k \in K} \# \text{Events}([u_k, u_{k+1})) \quad (\pm \text{ recorded discretization}).$$

Any rounding/discretization discrepancy is recorded as δ_{disc} (and, if measurement-induced, as δ_{meas}) in the run manifest (Appendix G).

Remark .72 (Alignment policy (after-collapse, same window)). When persistence and spectral measurements are combined, domain windows $\{[u_k, u_{k+1})\}$, collapse thresholds τ , and spectral bins must be *fixed per window* and logged. All B-side measurements are taken *after* applying \mathbf{T}_τ and on the same window; pre-collapse comparisons are out of scope. When definability is required (finite event sets, finite Čech depth), we assume Denef–Pas definability (Appendix Q), or the real o-minimal surrogate used in Appendices H/J.

Definition .73 (τ -adaptation, sweep, and stability bands). A threshold τ is *resolution-adapted* if $\tau = \alpha \cdot \max\{\Delta t, \Delta x\}$ for a fixed $\alpha > 0$. A τ -*sweep* is a discrete set $\{\tau_\ell\}$ on which diagnostics are evaluated. A *stability band* is a contiguous range $B \subset (0, \infty)$ such that the chosen natural transformations are isomorphisms for all $\tau \in B$ (hence the relevant tower diagnostics vanish on that band).

A.7. Clipping, strict commutation, and the δ -commutation contract

Let $\text{clip}_{[u,v]} : \text{Pers}_k^{\text{ft}} \rightarrow \text{Pers}_k^{\text{ft}}$ denote clipping to $[u, v]$ (half-open, right-open).

Lemma .74 (Strict commutation in the exact model). \mathbf{T}_τ commutes with clipping: $\mathbf{T}_\tau \circ \text{clip}_{[u,v]} \cong \text{clip}_{[u,v]} \circ \mathbf{T}_\tau$.

Proof. Clipping is exact and preserves interval lengths, hence preserves E_τ . It therefore descends to the Serre quotient and commutes with π_τ ; transporting across the equivalence in Theorem .64 gives the claim. \square

Proposition .75 (Operational δ -commutation). *In implementations that incur discretization/rounding, there exists a nonnegative defect $\delta_{\text{clip}}([u, v], \tau) \in V$ such that for all $M \in \text{Pers}_k^{\text{ft}}$*

$$d_V\left(\mathbf{T}_\tau \text{clip}_{[u,v]}(M), \text{clip}_{[u,v]} \mathbf{T}_\tau(M)\right) \leq \delta_{\text{clip}}([u, v], \tau),$$

with δ_{clip} aggregated by \oplus over concatenated windows and subadditive along pipelines. In the mathematical model, $\delta_{\text{clip}} = 0$ by Lemma .74.

A.8. V -enriched metric: benignity and operational role

Remark .76 (V -enrichment is benign; \mathbf{T}_τ as V -nucleus). Let V be a commutative quantale. The abelian/Serre-localization statements remain unchanged on the underlying category $\text{Pers}_k^{\text{ft}}$. The barcode semantics and length calculus (interval Jordan–Hölder and Serre quotient by E_τ) are invariant as *statements* in $\text{Pers}_k^{\text{ft}}$. The V -structure is used only to *measure* distances, sums, and gluing budgets uniformly; it does not alter the algebraic backbone. In particular, \mathbf{T}_τ is a V -nucleus (Prop. .56); operational non-commutations are summarized by δ -contracts (Remark .57, Prop. .75) and logged as part of the proof object (run.yaml; Appendix G).

A.9. Operational checklist and glued output

The reinforcement yields a single, self-contained persistence-layer backbone compatible with the v17.0 after-collapse contract:

- Abelianity and pointwise exactness (Proposition .60).
- Hereditary Serre τ -ephemeral class E_τ (Lemma .62); exact reflective localization (Theorem .64).
- \mathbf{T}_τ is an exact, idempotent, 1-Lipschitz V -nucleus (Corollary .65, Proposition .70, Proposition .56).
- Strict clipping-commutation in the categorical model (Lemma .74); operational δ -commutation contract for audits (Proposition .75).
- Shift-commutation and monotonicity in τ (Lemmas .68, .69); barcode-level description (Proposition .66).
- MECE windowing and coverage checks (Definition .71); τ -adaptation and stability bands (Definition .73).
- Filtered-colimit policy with return-to-constructible verification on declared windows (Remark .67), logged in the manifest (Appendix G).

Output: the globally glued object obtained from the windowed pipeline (MECE partition, localization, and subsequent certified steps in the main text) is accepted only under the mandated after-collapse order and under declared coverage/adaptation policies; all non-commutations are explicitly accounted for in the δ -ledger via the V -nucleus contracts above.

References for Appendix A. Crawley–Boevey (2015): Decomposition of pointwise finite-dimensional persistence modules. IMRN. Chazal–de Silva–Glisse–Oudot (2016): The Structure and Stability of Persistence Modules. Gabriel (1962): Des catégories abéliennes. Popescu: Abelian Categories. Stacks Project, Tag 02MO.

Appendix B. Lifting \mathbf{T}_τ to C_τ and the Homotopy Setting (reinforced v17.0)

Throughout, fix a field k . Let $\text{FiltCh}(k)$ denote the category of *bounded-in-degree* filtered chain complexes of finite-dimensional k -vector spaces with filtration-preserving chain maps. (“Bounded” refers to homological degree; filtrations are assumed *locally finite on bounded windows* as in Appendix A.) For each homological degree i , write

$$\mathbf{P}_i : \text{FiltCh}(k) \longrightarrow \text{Pers}_k^{\text{ft}}, \quad F \longmapsto (t \mapsto H_i(F^t C_\bullet)),$$

the degreewise persistence functor into the constructible subcategory (Appendix A).

Global scope and conventions (v17.0 alignment). (i) All claims at the filtered–complex layer hold *up to filtered quasi-isomorphism (f.q.i.)*; all identities at the persistence layer hold *strictly* in $\text{Pers}_k^{\text{ft}}$. (ii) Any operational quantitative comparison (distance, Lipschitz, monotonicity, budgets) is interpreted under the mandatory order

$$\boxed{\mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare in } \text{Pers}_k^{\text{ft}}}$$

on fixed right-open windows (after-collapse policy; Appendix A, Remark .54).

(iii) Filtered (co)limits, when invoked, are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$, and we then *verify* that the result lies in (or returns to) $\text{Pers}_k^{\text{ft}}$ on declared windows (Appendix A, Remark .67); no claim is made outside this regime.

(iv) Deletion-type updates are non-increasing for windowed energies and spectral tails *after truncation*, whereas inclusion-type updates are only stable (non-expansive) (Appendix E).

(v) Endpoint conventions follow Appendix A (Remark .63); in particular, infinite bars are not removed by \mathbf{T}_τ and their contributions are clipped by windowing.

(vi) For notational economy we sometimes write $\mathbf{T}_\tau = \mathbf{T}_\tau$.

(vii) *Amplitude guard-rail.* Realizations used operationally are taken with *amplitude* ≤ 1 *after collapse*: the fixed t -exact realization \mathcal{R} is required to have cohomological amplitude contained in $[0, 1]$ on the after-collapse objects $\mathcal{R}(C_\tau F)$, with this requirement enforced up to f.q.i. (§.87).

(viii) *δ -ledger interface.* Any implementation-level non-commutation/defect is charged into a commutative quantale $(V, \oplus, \leq, 0)$ as $\delta = \delta_{\text{alg}} \oplus \delta_{\text{disc}} \oplus \delta_{\text{meas}} \in V$ (Appendices A/L/S). Additivity along pipelines is always understood with respect to \oplus .

B.1. The interval-realization assignment \mathcal{U} (up to f.q.i.)

Definition .77 (Elementary interval blocks (two-term/one-term model)). Let $I[a, b]$ be an interval module (fixed endpoint convention; Appendix A, Remark .63).

- If $b < +\infty$, realize $I[a, b]$ in homological degree i by a *two-term filtered block*

$$k \cdot y \xrightarrow{d} k \cdot x, \quad |y| = i + 1, \quad |x| = i, \quad \text{fil}(x) = a, \quad \text{fil}(y) = b, \quad d(y) = x, \quad d(x) = 0.$$

Then x contributes a bar born at a and killed at b .

- If $b = +\infty$, realize $I[a, \infty)$ by a *one-term* block $k \cdot x$ in degree i with $\text{fil}(x) = a$ and $d = 0$.

In all blocks, the differential preserves the filtration: $d(F^t) \subseteq F^t$ for every t . Taking *locally finite on bounded windows* direct sums of such blocks and applying degree shifts produces a filtered complex whose persistence recovers the prescribed bars. We call any such model an *elementary interval complex* and denote a representative by $\mathcal{I}[a, b)$.

Proposition .78 (Barcode realization for bounded families (up to f.q.i.)). *There exists an assignment*

$$\mathcal{U} : \text{Pers}_k^{\text{ft}} \longrightarrow \text{FiltCh}(k)$$

such that for any degree-bounded family $\{M_i\}_{i \in \mathbb{Z}}$ of constructible persistence modules (only finitely many i nonzero) there are natural isomorphisms in $\text{Pers}_k^{\text{ft}}$,

$$\mathbf{P}_i\left(\bigoplus_j \mathcal{U}(M_j)[-j]\right) \cong M_i \quad (\forall i).$$

The construction is canonical up to filtered quasi-isomorphism, additive, and functorial in the homotopy category $\text{Ho}(\text{FiltCh}(k))$. In particular, for a single module M realized in a base degree (say 0) one has $\mathbf{P}_0(\mathcal{U}(M)) \cong M$ and $\mathbf{P}_j(\mathcal{U}(M)) = 0$ for all $j \neq 0$.

Remark .79 (Pseudofunctoriality of \mathcal{U}). The assignment \mathcal{U} extends to a *pseudofunctor* $\mathcal{U} : \text{Pers}_k^{\text{ft}} \rightarrow \text{Ho}(\text{FiltCh}(k))$: on a morphism of persistence modules, choose interval decompositions and a bar-matching; the induced blockwise filtered chain map is well-defined in Ho up to f.q.i., and compositions are respected up to coherent isomorphism. Consequently, constructions below that use \mathcal{U} on morphisms (e.g. C_τ) are functorial on $\text{Ho}(\text{FiltCh}(k))$.

B.2. Filtered quasi-isomorphisms and $\text{Ho}(\text{FiltCh}(k))$

Definition .80 (Filtered quasi-isomorphism). A filtration-preserving chain map $f : F \rightarrow G$ is a *filtered quasi-isomorphism (f.q.i.)* if for every $t \in \mathbb{R}$ the map $F^t C_\bullet \rightarrow G^t C_\bullet$ is a quasi-isomorphism. Equivalently, for all i , $\mathbf{P}_i(f)$ is an isomorphism in $\text{Pers}_k^{\text{ft}}$.

Lemma .81 (Characterization of f.q.i.). *For bounded-in-degree filtered complexes of finite-dimensional vector spaces, $f : F \rightarrow G$ is an f.q.i. iff $\mathbf{P}_i(f)$ is an isomorphism in $\text{Pers}_k^{\text{ft}}$ for all i .*

Definition .82 (Homotopy category). Let $\text{Ho}(\text{FiltCh}(k))$ be the localization of $\text{FiltCh}(k)$ at f.q.i.'s. Identities stated in $\text{Ho}(\text{FiltCh}(k))$ are to be understood *up to f.q.i.* at the model level. All endofunctors considered below (e.g. C_τ and Mirror/Transfer templates) preserve f.q.i.'s; thus they descend to $\text{Ho}(\text{FiltCh}(k))$.

B.2.1. Amplitude ≤ 1 : modeling caution and [Spec]

Remark .83 (Modeling caution: amplitude ≤ 1 after collapse). The operational guard-rail requires that, after applying C_τ , the realization $\mathcal{R}(C_\tau F)$ has cohomology concentrated in degrees $[0, 1]$. The *block-diagonal assembly* (no off-diagonal couplings across distinct bars or non-adjacent degrees) ensures this and is enforced throughout.

Declaration .84 ([Spec] amplitude ≤ 1 witness). Let $F \in \text{FiltCh}(k)$ and construct $C_\tau(F)$ by replacing each finite bar by a two-term block $(i+1) \rightarrow i$ and each infinite bar by a one-term block in degree i , with zero differentials between distinct blocks and between non-adjacent degrees. Then the spectral sequence of the stupid filtration on $\mathcal{R}(C_\tau F)$ degenerates at E_1 , and $H^q(\mathcal{R}(C_\tau F)) = 0$ for $q \notin \{0, 1\}$.

Proof sketch. Each bar contributes either a length-one complex or a single object; hence every differential raises degree by at most one. No off-diagonal maps exist by construction. Therefore the only potentially nonzero d_r occur with $r \leq 1$, so the spectral sequence collapses at E_1 , forcing cohomology to lie in degrees 0, 1. \square

B.2.2. f.q.i. checklist (T-Exactness-Persistence)

Definition .85 (Test T-Exactness-Persistence). Given $F \in \text{FiltCh}(k)$, the test consists of the following verifications:

- (E0) *Degree bound*: F is bounded in homological degree.
- (E1) *Local finiteness*: the filtration is locally finite on bounded windows.
- (E2) *Barcode audit*: for each i , $\mathbf{P}_i(F)$ is constructible (Appendix A).
- (E3) *Realization amplitude*: after C_τ , $\mathcal{R}(C_\tau F)$ has amplitude ≤ 1 (Declaration .84).
- (E4) *Exactness match*: for every short exact sequence of filtered complexes $0 \rightarrow F' \rightarrow F \rightarrow F'' \rightarrow 0$, the induced sequence of persistence modules is exact in $\text{Pers}_k^{\text{ft}}$.
- (E5) *Functoriality under f.q.i.*: if $f : F \rightarrow G$ is an f.q.i., then $\mathbf{P}_i(f)$ is iso for all i .
- (E6) *Shift/clip compatibility*: \mathbf{P}_i commutes with shifts and clipping; \mathbf{T}_τ commutes with clipping (Appendix A, Lemma .74).
- (E7) *Non-expansiveness (after-collapse)*:

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(F), \mathbf{T}_\tau \mathbf{P}_i(G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G)).$$

A dataset (F, τ) passes T-Exactness-Persistence if (E0)–(E7) hold.

Proposition .86 (Effect of T-Exactness-Persistence). *If F passes T-Exactness-Persistence, then for all i*

$$\mathbf{P}_i(C_\tau(F)) \cong \mathbf{T}_\tau \mathbf{P}_i(F),$$

functorially in $\text{Ho}(\text{FiltCh}(k))$. Moreover, C_τ preserves f.q.i. and is idempotent up to f.q.i.

Proof sketch. (E2) and (E4) place persistence computations in the abelian constructible regime where \mathbf{T}_τ is exact (Appendix A, Thm. .64); (E6) gives compatibility with clipping/shifts; (E7) is Appendix A, Prop. .70 applied under the mandated after-collapse order. The construction of C_τ (Theorem .87) together with (E3) ensures the amplitude guard-rail; f.q.i. detection by persistence (E5) yields the functorial identification in Ho . Idempotence follows from $\mathbf{T}_\tau \circ \mathbf{T}_\tau = \mathbf{T}_\tau$ and the block-diagonal lift. \square

B.3. Lifting \mathbf{T}_τ to C_τ and (co)limit/pullback compatibilities

Existence, functoriality, and uniqueness (homotopy-functor level): block-diagonal assembly.

Theorem .87 ([Declaration] Thresholded collapse in Ho). *For each $\tau \geq 0$ there exists an endofunctor*

$$C_\tau : \text{Ho}(\text{FiltCh}(k)) \longrightarrow \text{Ho}(\text{FiltCh}(k))$$

and natural isomorphisms in $\text{Pers}_k^{\text{ft}}$

$$\mathbf{P}_i(C_\tau(F)) \xrightarrow{\cong} \mathbf{T}_\tau(\mathbf{P}_i(F)) \quad (\forall i, F),$$

such that:

1. (Idempotence/monotonicity in Ho) $C_\tau \circ C_\sigma \simeq C_{\max\{\tau, \sigma\}} \simeq C_\sigma \circ C_\tau$.
2. (Non-expansiveness at persistence; after-collapse) $d_{\text{int}}(\mathbf{P}_i(C_\tau F), \mathbf{P}_i(C_\tau G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G))$.

Any two such lifts are uniquely isomorphic in $\text{Ho}(\text{FiltCh}(k))$. For $\tau = 0$, $C_0 \simeq \text{id}$ in $\text{Ho}(\text{FiltCh}(k))$.

Construction/Proof sketch. Replace $\mathbf{P}_i(F)$ with $\mathbf{T}_\tau(\mathbf{P}_i(F))$ (Appendix A) and realize via \mathcal{U} (Prop. .78); assemble differentials *block-diagonally* as in Remark .83. Functoriality and uniqueness follow from pseudo-functoriality of \mathcal{U} and the universal property of \mathbf{T}_τ ; non-expansiveness reflects Appendix A, Prop. .70. \square

(Co)limits and pullbacks: persistence layer is strict; filtered layer up to f.q.i.

Proposition .88 (Compatibility at the persistence layer). *Assume filtered colimits in $\text{FiltCh}(k)$ are computed degreewise and the results return to $\text{Pers}_k^{\text{ft}}$ on declared windows. Then for every filtered diagram $\{F_\lambda\}$ and every i ,*

$$\mathbf{P}_i(C_\tau(\varinjlim_\lambda F_\lambda)) \cong \varinjlim_\lambda \mathbf{P}_i(C_\tau(F_\lambda)) \quad \text{in } \text{Pers}_k^{\text{ft}}.$$

If, in addition, [Spec] finite pullbacks in $\text{FiltCh}(k)$ are computed degreewise and \mathcal{U} preserves finite limits up to f.q.i. under lifting-coherence ((LC)), then for any pullback square $F \times_H G$,

$$\mathbf{P}_i(C_\tau(F \times_H G)) \cong \mathbf{P}_i(C_\tau(F) \times_{C_\tau(H)} C_\tau(G)) \quad \text{in } \text{Pers}_k^{\text{ft}}.$$

At the filtered level, compatibilities hold up to f.q.i.

Remark .89 (On ((LC))). ((LC)) is a *finite-diagram* coherence ensuring that interval realizations can be chosen compatibly (up to f.q.i.) with pullback/pushout shapes encountered in practice. It holds for the block model and finite matching diagrams induced by monotone filtrations; we use it only in [Spec] statements.

Remark .90 (Realization functor; comparison maps [Spec]). Let $\mathcal{R} : \text{FiltCh}(k) \rightarrow D^b(k\text{-mod})$ be the fixed t -exact realization (amplitude guard-rail as above). Within the implementable range there are natural comparison morphisms

$$\mathcal{R} \circ C_\tau \implies \tau_{\geq 0} \circ \mathcal{R},$$

compatible with \mathbf{P}_i after homology (Appendix C). These maps are treated up to f.q.i. in $\text{Ho}(\text{FiltCh}(k))$.

B.4. Non-expansive Mirror/Transfer templates [Spec]

Definition .91 (Admissible Mirror/Transfer endofunctors). An endofunctor $\text{Mirror} : \text{FiltCh}(k) \rightarrow \text{FiltCh}(k)$ is *admissible* if:

1. (*Persistence non-expansiveness*) $d_{\text{int}}(\mathbf{P}_i(\text{Mirror } F), \mathbf{P}_i(\text{Mirror } G)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G))$ for all F, G, i .
2. (*Constructible stability*) Mirror carries finite-type objects to finite-type objects degreewise (on declared windows).
3. (*f.q.i.-invariance*) If f is an f.q.i., then $\text{Mirror}(f)$ is an f.q.i.; hence Mirror descends to $\text{Ho}(\text{FiltCh}(k))$.
4. (*Conditional commutation with C_τ*) There exists a natural 2-cell $\theta : \text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$ whose effect at persistence is δ -controlled:

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(\text{Mirror}(C_\tau F)), \mathbf{T}_\tau \mathbf{P}_i(C_\tau(\text{Mirror } F))) \leq \delta(i, \tau),$$

with $\delta(i, \tau)$ uniform in F and aggregated along pipelines via \oplus .

Theorem .92 (Quantitative commutation in Ho). *Assume Mirror is admissible and θ exists with bound $\delta(i, \tau)$. Then for all F, i, τ ,*

$$d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(\text{Mirror}(C_\tau F)), \mathbf{T}_\tau \mathbf{P}_i(C_\tau(\text{Mirror } F))\right) \leq \delta(i, \tau).$$

For a pipeline $\text{Mirror}_m, \dots, \text{Mirror}_1$ the bounds aggregate (in the quantale) as

$$d_{\text{int}}(\dots) \leq \delta_m(i, \tau_m) \oplus \dots \oplus \delta_1(i, \tau_1).$$

Any subsequent 1-Lipschitz persistence post-processing does not increase the bound.

B.5. Commutable torsion reflectors and A/B policy (homotopy interface)

Let $T_A, T_B : \text{Pers}_k^{\text{ft}} \rightarrow \text{Pers}_k^{\text{ft}}$ be exact reflectors with Serre classes E_A, E_B .

Proposition .93 (Nested torsions \Rightarrow order independence). *If $E_A \subseteq E_B$ or $E_B \subseteq E_A$, then $T_A \circ T_B = T_B \circ T_A = T_{A \vee B}$. In particular, for 1D length thresholds, $\mathbf{T}_\tau \circ \mathbf{T}_\sigma = \mathbf{T}_{\max\{\tau, \sigma\}}$.*

Definition .94 (A/B soft-commuting policy). For arbitrary reflectors T_A, T_B and $M \in \text{Pers}_k^{\text{ft}}$, set $\Delta_{\text{comm}}(M; A, B) := d_{\text{int}}(T_A T_B M, T_B T_A M)$. Given a tolerance $\eta \geq 0$, accept *soft-commuting* if $\Delta_{\text{comm}} \leq \eta$; otherwise fix an order and charge Δ_{comm} to δ^{alg} (Appendix L) with pipeline aggregation via \oplus .

B.6. AWFS on Ho(FiltCh) and 2-cell accounting

Declaration .95 (AWFS on Ho(FiltCh)). We adopt an algebraic weak factorization system on Ho(FiltCh) with L (preprocess/left map) and R (collapse/right map) such that $R \simeq C_\tau$ up to f.q.i. Triangle/zigzag identities hold up to f.q.i.; all 2-cell deviations are recorded as δ_{alg} (Appendix L).

Theorem .96 (AWFS triangle 2-cells). *There exist coherent 2-cells (quantitatively bounded after collapse)*

$$C_\tau \circ C_\tau \simeq C_\tau, \quad \mathbf{T}_\tau \circ \mathbf{T}_\tau = \mathbf{T}_\tau, \quad L \circ R \simeq R \circ L,$$

whose implementation non-commutation is absorbed into the δ_{alg} -budget (Appendix L).

Corollary .97 (A/B policy as 2-cell accounting). *For two exact reflectors on persistence, any measured defect Δ_{comm} (Definition .94) is realizable as a 2-cell deviation within the AWFS picture and must be logged as δ_{alg} ; pipeline aggregation follows from the quantale-sum rule in Appendix L.*

Remark .98 (V-shifts and C_τ in situ). If S^v denotes a Lawvere V -shift compatible with degree-wise filtrations, then $C_\tau \circ S^v \simeq S^v \circ C_\tau$ in Ho and, after applying \mathbf{P}_i and then \mathbf{T}_τ , the induced comparison is V -1-Lipschitz (Appendix A, Lemma .68).

B.7. Completion note and implementation recipe

Remark .99 (No further supplementation required). This appendix integrates: (i) the lift C_τ of the exact reflector \mathbf{T}_τ to the homotopy setting (existence, functoriality, uniqueness up to f.q.i.; non-expansiveness at persistence); (ii) strict persistence-layer compatibilities with (co)limits and pullbacks (filtered level up to f.q.i.); (iii) admissible Mirror/Transfer templates with a uniform, additive 2-cell bound $\delta(i, \tau)$ stable under 1-Lipschitz post-processing; (iv) a commutable-torsion policy (nested \Rightarrow order-independent; else A/B with ledger); and (v) *the amplitude ≤ 1 guard-rail and the f.q.i. checklist* T-Exactness-Persistence. No additional supplementation is required for operational use under the global after-collapse scope of Appendix A.

Implementation recipe (engineering checklist).

- Build C_τ by block-diagonal assembly of interval blocks; forbid off-diagonal couplings across blocks/degrees (enforces amplitude ≤ 1).
- For each morphism, lift $\mathbf{T}_\tau \mathbf{P}_i(f)$ blockwise via \mathcal{U} and take direct sums across i ; functorial in Ho.
- Enforce T-Exactness-Persistence (Def. .85); expose pass/fail and witnesses in the manifest/log layer.
- For Mirror/Transfer, provide a 2-cell θ with a *uniform* $\delta(i, \tau)$; aggregate along pipelines via \oplus ; ensure post-processors are 1-Lipschitz at persistence (after-collapse).
- For reflectors, run A/B tests; if $\Delta_{\text{comm}} > \eta$, fix an order and charge the surplus to δ^{alg} (Appendix L).
- Keep clipping/windowing separate from localization; use Appendix A, Lemma .74 as needed.

References for Appendix B. Crawley–Boevey (2015): Decomposition of pointwise finite-dimensional persistence modules. IMRN. Chazal–de Silva–Glisse–Oudot (2016): The Structure and Stability of Persistence Modules. Standard sources on AWFS and homotopical algebra (e.g. Riehl, *Categorical Homotopy Theory*) are used at the level of *up to f.q.i.* coherence only.

Appendix C. The Bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ and its Local Reverse under $E_1=0$ (reinforced, complete v17.0)

Throughout, fix a field k . Let $\text{FiltCh}(k)$ be the category of *bounded-in-degree* filtered chain complexes of finite-dimensional k -vector spaces with filtration-preserving maps. For $F \in \text{FiltCh}(k)$ and each degree i , the degreewise persistence functor

$$\mathbf{P}_i(F) : \mathbb{R} \longrightarrow \text{Vect}_k, \quad t \longmapsto H_i(F^t C_\bullet)$$

is assumed *constructible* (pointwise finite-dimensional, finitely many critical parameters on bounded windows), i.e. $\mathbf{P}_i(F) \in \text{Pers}_k^{\text{ft}}$ (Appendix A). We also fix a realization functor

$$\mathcal{R} : \text{FiltCh}(k) \longrightarrow D^b(k\text{-mod})$$

into the bounded derived category of finite-dimensional k -vector spaces.

Bridge hypotheses, scope, and gate policy (v17.0 canon). We work under the standing assumptions (B1)–(B3) used in the main text:

- (B1) field coefficients and constructibility of all $\mathbf{P}_i(F)$ on declared windows;
- (B2) *two-term (amplitude ≤ 1) guard-rail after collapse*: on any window where the Ext^1 -test is *eligible*, the after-collapse object satisfies

$$\mathcal{R}(C_\tau F) \in D^{[-1,0]}(k\text{-mod})$$

(equivalently: homological amplitude ≤ 1 under the chain/derived sign convention);

- (B3) functoriality/naturality of all constructions (including clipping, collapse, and realization) up to f.q.i. where applicable.

All statements in this appendix are confined to the implementable range of Appendix A and the homotopy interface of Appendix B.

Eligibility for B-Gate⁺ (fixed). The Ext^1 -test is included in B-Gate⁺ *only* on windows/scales where $\mathcal{R}(C_\tau F) \in D^{[-1,0]}(k\text{-mod})$ and the test object is $k[0]$. Outside this regime, Ext^1 may be logged but is *not* used for gating.

Operational order (collapse-first; fixed). All gate-relevant categorical tests follow the mandated order

$$\boxed{F \xrightarrow{C_\tau} C_\tau F \xrightarrow{\text{clip}_W} (C_\tau F)|_W \xrightarrow{\mathcal{R}} \mathcal{R}((C_\tau F)|_W) \xrightarrow{\text{Ext}^1(-, k[0])} 0}.$$

This is compatible with the persistence-layer canon $\mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare}$, via Appendix B: $\mathbf{P}_i(C_\tau F) \cong \mathbf{T}_\tau(\mathbf{P}_i(F))$.

Meaning of $\text{PH}_1 = 0$. For a filtered complex F , we write

$$\text{PH}_1(F) = 0 \quad :\Longleftrightarrow \quad \mathbf{P}_1(F) = 0 \text{ in } \text{Pers}_k^{\text{ft}} \quad \Longleftrightarrow \quad H_1(F^t) = 0 \quad \forall t \in \mathbb{R}.$$

All operational uses are on after-collapse/windowed objects, i.e. $\text{PH}_1((C_\tau F)|_W) = 0$.

Windows, clipping, and commutation. A *right-open window* is a half-open interval $W = [u, v)$ (right endpoint open), consistent with the global MECE/window convention (Appendix A, §A.6). We write clip_W for clipping/restriction to W (Appendix A, §A.7). In the exact model, \mathbf{T}_τ commutes strictly with clipping (Appendix A, Lemma .74); any implementation defect is charged as δ_{clip} (Appendix A, Proposition .75).

C.1. Two-term amplitude and a canonical two-term model

Proposition .100 (Two-term amplitude normal form). *If $A \in D^{[-1,0]}(k\text{-mod})$, then there is a natural isomorphism in $D^b(k\text{-mod})$*

$$A \simeq \left[H^{-1}(A) \xrightarrow{d_A} H^0(A) \right],$$

where the complex on the right is concentrated in cohomological degrees -1 and 0 .

Remark .101. All statements below are invariant under filtered quasi-isomorphism on F and under isomorphism in $D^b(k\text{-mod})$ on $\mathcal{R}(F)$.

C.2. The edge: $H^{-1}(\mathcal{R}(F)) \cong \varinjlim_t H_1(F^t)$

Assumption .102 (Edge-compatibility of the realization). *On any eligible window (i.e. where $\mathcal{R}(C_\tau F) \in D^{[-1,0]}$), the realization \mathcal{R} is assumed to be compatible with filtered colimits along the filtration parameter in the following sense:*

$$H^{-1}(\mathcal{R}(F)) \cong \varinjlim_{t \in \mathbb{R}} H_1(F^t C_\bullet),$$

naturally in F (and similarly after clipping to a fixed right-open window W by restricting the directed system to $t \in W$). This is the only place where a compatibility between \mathcal{R} and the filtration-directed system is used.

Proposition .103 (Edge identification and naturality). *Assume (B1)–(B3) and Assumption .102. Then for every eligible $F \in \text{FiltCh}(k)$ there is a natural isomorphism*

$$H^{-1}(\mathcal{R}(F)) \cong \varinjlim_{t \in \mathbb{R}} H_1(F^t C_\bullet).$$

If $f : F \rightarrow G$ preserves filtrations, the induced square with the maps on colimits commutes. The same statement holds after applying C_τ and/or clipping to a right-open window W .

C.3. Computing Ext^1 in amplitude $[-1, 0]$

Lemma .104 (Edge lemma for Ext^1 in amplitude $[-1, 0]$). *For $A \in D^{[-1, 0]}(k\text{-mod})$ there is a natural isomorphism*

$$\text{Ext}^1(A, k[0]) \cong \text{Hom}_k(H^{-1}(A), k).$$

Proof sketch. Use the standard truncation triangle $\tau_{\leq -1}A \rightarrow A \rightarrow \tau_{\geq 0}A \rightarrow$, where $\tau_{\leq -1}A \simeq H^{-1}(A)[1]$ and $\tau_{\geq 0}A \simeq H^0(A)[0]$. Apply $\text{RHom}(-, k[0])$ and take H^1 , using $\text{Ext}^{>0}(V, k) = 0$ for finite-dimensional k -spaces V . \square

Corollary .105 (Dimension and perfect pairing). *For any eligible F (so $\mathcal{R}(F) \in D^{[-1, 0]}$),*

$$\dim_k \text{Ext}^1(\mathcal{R}(F), k[0]) = \dim_k H^{-1}(\mathcal{R}(F)) = \dim_k \left(\varinjlim_t H_1(F^t) \right),$$

and the canonical pairing $\text{Ext}^1(\mathcal{R}(F), k[0]) \otimes H^{-1}(\mathcal{R}(F)) \rightarrow k$ is perfect.

C.4. The forward bridge and its after-collapse/windowed form

Theorem .106 (Bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ (eligible regime)). *Let $F \in \text{FiltCh}(k)$ be eligible (so $\mathcal{R}(F) \in D^{[-1, 0]}$). If $\text{PH}_1(F) = 0$ (equivalently, $H_1(F^t) = 0$ for all t), then*

$$\text{Ext}^1(\mathcal{R}(F), k[0]) = 0.$$

Proof. If $\text{PH}_1(F) = 0$, then $\varinjlim_t H_1(F^t) = 0$. By Proposition .103, $H^{-1}(\mathcal{R}(F)) = 0$. Apply Lemma .104. \square

Corollary .107 (After-collapse/windowed bridge (gate form)). *Fix $\tau > 0$ and a right-open window $W = [u, v)$. Assume eligibility on W , i.e. $\mathcal{R}((C_\tau F)|_W) \in D^{[-1, 0]}$. If $\text{PH}_1((C_\tau F)|_W) = 0$, then*

$$\text{Ext}^1(\mathcal{R}((C_\tau F)|_W), k[0]) = 0.$$

C.5. The local reverse under $E_1=0$ (formal statement of P3)

Definition .108 (Tail stabilization on a right-open window). Let $W = [u, v)$ be right-open. We say that a persistence module $M \in \text{Pers}_k^{\text{ft}}$ has *tail stabilization on W* if there exists $t_0 \in (u, v)$ such that for all $t_0 \leq t \leq t' < v$, the structure map $M(t) \rightarrow M(t')$ is an isomorphism. We say $F \in \text{FiltCh}(k)$ has tail stabilization on W in degree 1 if $\mathbf{P}_1(F|_W)$ has tail stabilization.

Remark .109 (About $E_1(W) = 0$). The predicate $E_1(W) = 0$ is the window-local trigger defined in Chapter 11 (energy/spectral layer) and is evaluated *after collapse* on W . In this appendix we only use that $E_1(W) = 0$, together with tail stabilization, identifies the stabilized degree-1 edge on W with the windowed persistent H_1 in the eligible regime.

Lemma .110 (Edge identification on E_1 -degenerate windows). *Let $W = [u, v)$ be right-open. Assume $E_1(W) = 0$ and that $(C_\tau F)$ has tail stabilization on W in degree 1. Assume also eligibility on W : $\mathcal{R}((C_\tau F)|_W) \in D^{[-1, 0]}$. Then there is a natural isomorphism*

$$H^{-1}(\mathcal{R}((C_\tau F)|_W)) \cong \text{PH}_1((C_\tau F)|_W).$$

Proof sketch. Tail stabilization on W identifies the directed colimit $\lim_{t \in W} H_1(((C_\tau F)|_W)^t)$ with the stabilized value of the degree-1 persistence on W . The hypothesis $E_1(W) = 0$ provides the E_1 -degeneracy needed to identify this stabilized edge with the windowed persistent class used by PH_1 in the gate layer (Chapter 11). Combine with the edge compatibility in Proposition .103. \square

Theorem .111 (Local Reverse under $E_1=0$ (P3; eligible regime)). *Let $W = [u, v]$ be right-open. Assume $E_1(W) = 0$, tail stabilization on W for $(C_\tau F)$ in degree 1, and eligibility on W : $\mathcal{R}((C_\tau F)|_W) \in D^{[-1,0]}$. If*

$$\text{Ext}^1(\mathcal{R}((C_\tau F)|_W), k[0]) = 0,$$

then

$$\text{PH}_1((C_\tau F)|_W) = 0.$$

Proof. By Lemma .110, $H^{-1}(\mathcal{R}((C_\tau F)|_W)) \cong \text{PH}_1((C_\tau F)|_W)$. By Lemma .104, $\text{Ext}^1(-, k[0]) \cong \text{Hom}(H^{-1}(-), k)$ in the eligible regime. Hence $\text{Ext}^1 = 0 \Rightarrow H^{-1} = 0 \Rightarrow \text{PH}_1 = 0$. \square

Corollary .112 (Local equivalence on suitable windows). *Let $W = [u, v]$ be right-open and (UCC-)definable with finite event set and finite Čech depth (Appendix Q; cf. Appendix A for window conventions). Assume eligibility on W and tail stabilization on W for $(C_\tau F)$ in degree 1. If $E_1(W) = 0$, then for every F and $\tau > 0$,*

$$\text{PH}_1((C_\tau F)|_W) = 0 \iff \text{Ext}^1(\mathcal{R}((C_\tau F)|_W), k[0]) = 0,$$

with \Rightarrow by Theorem .106 and \Leftarrow by Theorem .111.

C.6. Naturality, exact triangles, and stability

Proposition .113 (Naturality of the Ext^1 edge isomorphism). *For any morphism $f : F \rightarrow G$ preserving filtrations, under eligibility the natural square*

$$\begin{array}{ccc} \text{Ext}^1(\mathcal{R}(F), k[0]) & \xrightarrow{\sim} & \text{Hom}(H^{-1}(\mathcal{R}(F)), k) \\ \text{Ext}^1(\mathcal{R}(f), k[0]) \downarrow & & \downarrow \text{Hom}(H^{-1}(\mathcal{R}(f)), k) \\ \text{Ext}^1(\mathcal{R}(G), k[0]) & \xrightarrow{\sim} & \text{Hom}(H^{-1}(\mathcal{R}(G)), k) \end{array}$$

commutes.

Lemma .114 (Exact triangles and 2-out-of-3 for Ext^1). *Let $A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_1[1]$ be a distinguished triangle in $D^b(k\text{-mod})$ with each $A_i \in D^{[-1,0]}$. If two of $\text{Ext}^1(A_i, k[0])$ vanish, then so does the third.*

Remark .115 (Stability under admissible updates (eligible regime)). *If $F \mapsto F'$ is an admissible update that is non-expansive at the persistence layer after collapse (e.g. deletion-type updates), and both $\mathcal{R}(C_\tau F)$ and $\mathcal{R}(C_\tau F')$ are eligible, then the verdict $\text{Ext}^1(\mathcal{R}(C_\tau F), k[0]) = 0$ is stable under $F \mapsto F'$ whenever the stabilized edge group (equivalently H^{-1}) is preserved.*

C.7. Implementation details, reproducibility, and logging

Remark .116 (Run-time policy and manifest fields). *Enforce the order $F \rightarrow C_\tau F \rightarrow (C_\tau F)|_W \rightarrow \mathcal{R}((C_\tau F)|_W) \rightarrow \text{Ext}^1(-, k[0])$. The manifest run.yaml must include (per window W and τ):*

- `ext1_eligible`: boolean; true iff $\mathcal{R}((C_\tau F)|_W) \in D^{[-1,0]}$;
- `ext1_used_in_gate`: boolean; true iff `ext1_eligible` and gate policy enables it;

- amplitude: reported as " $[-1,0]$ " or " >1 ";
- gate_order: "collapse→clip→realize→ext1";
- q_test: "k[0]";
- window_E1_zero: boolean (Chapter 11 predicate, evaluated after collapse);
- tail_stable_H1: boolean (Definition .108);
- notes: free text; if window_E1_zero:true and tail_stable_H1:true, then "Local reverse (P3) enabled".

Outside eligibility, set ext1_eligible:false, ext1_used_in_gate:false, and continue the gate using persistence/tower/safety criteria only.

```
# run.yaml (excerpt; per window W and tau)
ext1_eligible: true
ext1_used_in_gate: true
amplitude: "[-1,0]"
gate_order: "collapse→clip→realize→ext1"
q_test: "k[0]"
window_E1_zero: true
tail_stable_H1: true
notes: "Local reverse (P3) enabled on E1=0 window with tail stabilization."
```

C.8. Counterexamples and boundary cases

Example .117 (Failure of the global reverse implication $\text{Ext}^1 \Rightarrow \text{PH}_1$). A single *finite* degree-1 bar yields $\text{PH}_1(F) \neq 0$ while the directed colimit $\lim_{t \rightarrow +\infty} H_1(F^t) = 0$, hence $\text{Ext}^1(\mathcal{R}(F), k[0]) = 0$ in the eligible regime. Thus the converse fails globally (and this is why the reverse direction is only asserted locally under $E_1(W) = 0$ with tail stabilization).

Example .118 (Amplitude breach: diagnostics only). If $\mathcal{R}((C_\tau F)|_W) \notin D^{[-1,0]}$ (e.g. it lies in $D^{[-2,0]}$ with $H^{-2} \neq 0$), then Lemma .104 is inapplicable. The Ext^1 -test is *ineligible* and must not be used for gating.

Example .119 (Non-constructible tails). If $\mathbf{P}_1(F) \notin \text{Pers}_k^{\text{ft}}$, filtered colimits in $[\mathbb{R}, \text{Vect}_k]$ may exit the constructible regime (Appendix A, Remark .67). Eligibility fails by (B1).

C.9. Additional safeguards and best practices (policy layer)

Remark .120 (Monotonicity across thresholds (implementation policy)). In the block-diagonal reference implementation (Appendix B), eligibility is monotone under strengthening collapse: if $\mathcal{R}((C_{\tau'} F)|_W) \in D^{[-1,0]}$ for some $\tau' \geq \tau$, then $\mathcal{R}((C_\tau F)|_W) \in D^{[-1,0]}$. Accordingly, eligibility is logged per (W, τ) , but the implementation may reuse witnesses across a τ -sweep when monotonicity is verified.

Remark .121 (Uniformity under base change (benignity)). If $k \subset K$ is a field extension, then (within the eligible regime) the vanishing verdict $\text{Ext}^1(\mathcal{R}((C_\tau F)|_W), k[0]) = 0$ is stable under scalar extension to K . Thus the gate decision is field-independent within the amplitude $[-1, 0]$ regime.

C.10. Scope and non-claims

Remark .122 (Scope and non-claims). The forward bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ is asserted and used only in the eligible regime $\mathcal{R}((C_\tau F)|_W) \in D^{[-1,0]}$, and only for the test object $k[0]$. The global converse is *false*. The *local reverse* (Theorem .111, formal statement of P3) holds on right-open windows where $E_1(W) = 0$ and the degree-1 tail stabilizes, in the eligible regime. All uses respect the after-collapse order and the filtered-colimit scope policy of Appendix A. No claims are made outside the persistence/derived layers in the implementable range.

Summary of Appendix C (reinforced and complete). Under (B1)–(B3) and eligibility $(\mathcal{R}((C_\tau F)|_W) \in D^{[-1,0]})$, the edge compatibility identifies $H^{-1}(\mathcal{R}((C_\tau F)|_W))$ with the stabilized windowed H_1 , and the amplitude $[-1, 0]$ calculus yields $\text{Ext}^1(\mathcal{R}((C_\tau F)|_W), k[0]) \cong \text{Hom}(H^{-1}(\cdot), k)$. This gives the forward bridge $\text{PH}_1((C_\tau F)|_W) = 0 \Rightarrow \text{Ext}^1(\mathcal{R}((C_\tau F)|_W), k[0]) = 0$, and, on $E_1(W) = 0$ windows with tail stabilization, the local reverse $\text{Ext}^1 = 0 \Rightarrow \text{PH}_1 = 0$ (P3). Operationally, the order is **collapse** \rightarrow **clip** \rightarrow **realize** \rightarrow **Ext**; outside eligibility, Ext^1 is excluded from gate decisions.

Appendix D. Towers, μ , u , and Examples [Proof/Example] (reinforced, complete v17.0)

Throughout, fix a field k . We work in the constructible regime (Appendix A). Let $\text{FiltCh}(k)$ denote the category of bounded-in-degree filtered chain complexes of finite-dimensional k -vector spaces with filtration-preserving chain maps (Appendix B). For each degree $i \in \mathbb{Z}$, the degreewise persistence functor is

$$\mathbf{P}_i : \text{FiltCh}(k) \longrightarrow \text{Pers}_k^{\text{ft}}, \quad F \longmapsto (t \mapsto H_i(F^t C_\bullet)).$$

The truncation/deletion reflector $\mathbf{T}_\tau : \text{Pers}_k^{\text{ft}} \rightarrow \text{Pers}_{k, \tau\text{-loc}}^{\text{ft}}$ is exact and 1-Lipschitz (Appendix A). Filtered colimits are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$, subject to the scope rule of Appendix A, Remark .67. All statements at the filtered–complex layer are *up to filtered quasi-isomorphism (f.q.i.)*; persistence-layer statements take place *strictly* in $\text{Pers}_k^{\text{ft}}$. All quantities below may depend on the threshold $\tau > 0$; *no monotonicity in τ* is asserted.

Notation for “Type IV” verdicts at scale τ . For a fixed tower and threshold τ , write

$$(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau) \iff \mu_{i, \tau} = 0 \text{ and } u_{i, \tau} = 0 \text{ for all relevant } i,$$

and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \neq (0, 0)(\tau)$ otherwise. This is the *tower-level (Type IV) defect detector* at scale τ .

Proposition A.1 (Calculus of μ , u in the V -metric). *Let V be a commutative quantale and endow $\text{Pers}_k^{\text{ft}}$ with a Lawvere V -metric (Chapter 2; Appendix A, Remark .76). For any tower with apex and any $\tau > 0$, the obstruction indices computed after \mathbf{T}_τ ,*

$$\mu_{i, \tau} := \text{gdim Ker}(\phi_{i, \tau}), \quad u_{i, \tau} := \text{gdim coker}(\phi_{i, \tau}),$$

satisfy (uniformly in the choice of V):

1. **Cofinal/f.q.i. invariance:** μ , u are invariant under cofinal reindexing of the tower and under levelwise f.q.i. replacements.
2. **Composition subadditivity:** for composable comparison maps ψ, ϕ ,

$$\mu(\psi \circ \phi) \leq \mu(\phi) + \mu(\psi), \quad u(\psi \circ \phi) \leq u(\psi) + u(\phi).$$

$$\begin{array}{ccccccc}
F_1 & \longrightarrow & F_2 & \longrightarrow & \cdots & \longrightarrow & F_n & \longrightarrow & \cdots & \longrightarrow & F_\infty \\
& \searrow & & \searrow & & & \searrow & & & & \\
& & \mathbf{P}_i(F_1) & \longrightarrow & \mathbf{P}_i(F_2) & \longrightarrow & \cdots & \longrightarrow & \mathbf{P}_i(F_n) & \longrightarrow & \mathbf{P}_i(F_\infty)
\end{array}$$

Figure 3: A tower with apex F_∞ and its image under \mathbf{P}_i . The comparison $\phi_{i,\tau}$ (defined after applying \mathbf{T}_τ) measures the failure of the cocone to exhibit a colimit at scale τ .

3. **Additivity on finite sums:** μ , u are additive under finite direct sums of towers.

4. **Window finiteness on definable covers:** on o -minimal definable windows with finite Čech/Leray depth (Appendix H/J), all $\mu_{i,\tau}$, $u_{i,\tau}$ are finite and only finitely many τ -criticalities occur per window.

All multiplicities are read as the number of $I[0, \infty)$ summands in the barcode of the relevant kernel/cokernel after \mathbf{T}_τ .

Proof sketch. By Appendix A, Remark .76, V -enrichment does not alter the underlying abelian/Serre calculus: kernels, cokernels, and \mathbf{T}_τ are computed in $\text{Pers}_k^{\text{ft}}$ as in the unenriched case. Hence (1)–(3) follow from the barcode calculus and the linear-algebra inequalities used below (Propositions A.4, A.6, A.7). For (4), definable covers yield finite depth, so only finitely many events contribute on a window (Appendix H/J), and finiteness follows. \square

Comparison map and obstruction indices. Let $\{F_n\}_{n \in \mathbb{N}}$ be a directed system in $\text{FiltCh}(k)$. Let F_∞ be an apex equipped with a cocone $F_n \rightarrow F_\infty$ (indexing category $\mathbb{N} \cup \{\infty\}$ with unique morphisms $n \rightarrow \infty$). For each i and $\tau > 0$ define the comparison morphism in $[\mathbb{R}, \text{Vect}_k]$

$$\phi_{i,\tau} : \text{colim}_{n \in \mathbb{N}} \mathbf{T}_\tau(\mathbf{P}_i(F_n)) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(F_\infty)).$$

Define the *tower obstruction indices*

$$\mu_{i,\tau} := \text{gdim Ker}(\phi_{i,\tau}), \quad u_{i,\tau} := \text{gdim coker}(\phi_{i,\tau}), \quad \mu_{\text{tot},\tau} := \sum_i \mu_{i,\tau}, \quad u_{\text{tot},\tau} := \sum_i u_{i,\tau}.$$

These sums are finite because complexes are bounded in homological degrees and we work in the constructible range.

Remark A.2 (Generic dimension after truncation). In $\text{Pers}_k^{\text{ft}}$, after applying \mathbf{T}_τ the kernel and cokernel of any morphism decompose (noncanonically) as finite direct sums of interval modules. We write $\text{gdim}(-)$ for the *generic fiber dimension*, i.e. the multiplicity of the infinite bar $I[0, \infty)$ in that decomposition. Finite bars contribute zero generic fiber.

Remark A.3 (Invariance of $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)/(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$). The indices (μ, u) (hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)/(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$) are invariant under levelwise f.q.i. replacements of the tower and apex: if $F_n \simeq_{\text{f.q.i.}} F'_n$ and $F_\infty \simeq_{\text{f.q.i.}} F'_\infty$, then \mathbf{P}_i sends these to isomorphisms in $\text{Pers}_k^{\text{ft}}$, hence kernels/cokernels (and thus μ, u) are unchanged. They are also invariant under cofinal reindexing of the tower, since filtered colimits over cofinal subdiagrams are canonically isomorphic.

D.1. Calculus of defects: generic-fiber interpretation, naturality, and subadditivity

Proposition A.4 (Generic-fiber interpretation). *Let $f : M \rightarrow N$ be a morphism in $\text{Pers}_k^{\text{ft}}$ and fix $\tau > 0$. Then, in the barcode decomposition of $\text{Ker}(\mathbf{T}_\tau f)$ and $\text{coker}(\mathbf{T}_\tau f)$, the multiplicity of $I[0, \infty)$ equals $\text{gdim Ker}(\mathbf{T}_\tau f)$ and $\text{gdim coker}(\mathbf{T}_\tau f)$, respectively. Equivalently, gdim is the generic fiber dimension of the corresponding functor $\mathbb{R} \rightarrow \text{Vect}_k$.*

Proof. Standard for constructible 1D persistence over a field: after \mathbf{T}_τ , objects split as finite sums of interval modules (Appendix A). The generic fiber (dimension on a cofinal ray) counts infinite bars, i.e. copies of $I[0, \infty)$. \square

Definition A.5 (Morphisms of towers and naturality of ϕ). A *morphism of towers with apex* $(F_\bullet, F_\infty) \rightarrow (G_\bullet, G_\infty)$ is a collection of maps $u_n : F_n \rightarrow G_n$ and $u_\infty : F_\infty \rightarrow G_\infty$ commuting with all structure maps to the apex. Applying \mathbf{P}_i , then \mathbf{T}_τ , and passing to the filtered colimit yields a commutative square

$$\begin{array}{ccc} \operatorname{colim}_n \mathbf{T}_\tau \mathbf{P}_i(F_n) & \xrightarrow{\phi_{i,\tau}^F} & \mathbf{T}_\tau \mathbf{P}_i(F_\infty) \\ \operatorname{colim} \mathbf{T}_\tau \mathbf{P}_i(u_n) \downarrow & & \downarrow \mathbf{T}_\tau \mathbf{P}_i(u_\infty) \\ \operatorname{colim}_n \mathbf{T}_\tau \mathbf{P}_i(G_n) & \xrightarrow{\phi_{i,\tau}^G} & \mathbf{T}_\tau \mathbf{P}_i(G_\infty), \end{array}$$

i.e. $\phi_{i,\tau}$ is natural in the tower.

Proposition A.6 (Subadditivity under composition). *Let $(F_\bullet, F_\infty) \xrightarrow{u} (G_\bullet, G_\infty) \xrightarrow{v} (H_\bullet, H_\infty)$ be morphisms of towers with apex. Then, for each i, τ ,*

$$\mu(\phi_{i,\tau}^H \circ \operatorname{colim} \mathbf{T}_\tau \mathbf{P}_i(v_n \circ u_n)) \leq \mu(\phi_{i,\tau}^G \circ \operatorname{colim} \mathbf{T}_\tau \mathbf{P}_i(u_n)) + \mu(\phi_{i,\tau}^H \circ \operatorname{colim} \mathbf{T}_\tau \mathbf{P}_i(v_n)),$$

$$u(\phi_{i,\tau}^H \circ \operatorname{colim} \mathbf{T}_\tau \mathbf{P}_i(v_n \circ u_n)) \leq u(\phi_{i,\tau}^H \circ \operatorname{colim} \mathbf{T}_\tau \mathbf{P}_i(v_n)) + \mu(\phi_{i,\tau}^G \circ \operatorname{colim} \mathbf{T}_\tau \mathbf{P}_i(u_n)).$$

In particular, for any factorization of a fixed comparison map ϕ , one has $\mu(\psi \circ \phi) \leq \mu(\phi) + \mu(\psi)$ and $u(\psi \circ \phi) \leq u(\psi) + \mu(\phi)$.

Proof. Apply \mathbf{T}_τ and use exactness together with the standard inequalities for kernels and cokernels of compositions in finite-dimensional linear algebra, then pass to generic fibers via Proposition A.4. \square

Proposition A.7 (Additivity on finite direct sums). *For two towers (F_\bullet, F_∞) and (G_\bullet, G_∞) ,*

$$\mu((F \oplus G)_\bullet, (F \oplus G)_\infty) = \mu(F_\bullet, F_\infty) + \mu(G_\bullet, G_\infty), \quad u((F \oplus G)_\bullet, (F \oplus G)_\infty) = u(F_\bullet, F_\infty) + u(G_\bullet, G_\infty).$$

Proof. \mathbf{P}_i and \mathbf{T}_τ preserve finite direct sums; kernels/cokernels preserve finite direct sums; gdim is additive on direct sums. \square

Proposition A.8 (Invariance under f.q.i. and cofinal reindexing). *If $F_n \simeq_{\text{f.q.i.}} F'_n$ levelwise and $F_\infty \simeq_{\text{f.q.i.}} F'_\infty$, then μ, u agree for the two towers. If $J \subset \mathbb{N}$ is cofinal, then restricting the tower to J does not change μ, u .*

Proof. \mathbf{P}_i sends f.q.i. maps to isomorphisms in $\operatorname{Pers}_k^{\text{ft}}$; \mathbf{T}_τ is exact; filtered colimits over cofinal subdiagrams are canonically isomorphic. \square

D.2. Toy towers: pure kernel / pure cokernel / mixed

Example A.9 (Pure cokernel at a fixed scale). Fix $\tau > 0$ and degree $i = 1$. Let $\mathbf{P}_1(F_n) = I[0, \tau - \frac{1}{n})$ with transition maps the evident inclusions. Let F_∞ satisfy $\mathbf{P}_1(F_\infty) = I[0, \infty)$. Then $\mathbf{T}_\tau(\mathbf{P}_1(F_n)) = 0$ for all n , whereas $\mathbf{T}_\tau(\mathbf{P}_1(F_\infty)) \cong I[0, \infty)$. Hence $\phi_{1,\tau} : 0 \rightarrow I[0, \infty)$ has trivial kernel and nontrivial cokernel, so $\mu_{1,\tau} = 0$ and $u_{1,\tau} = 1$ (pure cokernel).

Example A.10 (Pure kernel at a fixed scale). Fix $\tau > 0$. Let $\mathbf{P}_1(F_n) = I[0, \infty)$ for all n , with transition maps the identities. Let F_∞ satisfy $\mathbf{P}_1(F_\infty) = 0$, and take the cocone maps $\mathbf{P}_1(F_n) \rightarrow \mathbf{P}_1(F_\infty)$ to be 0 for all n . Then $\mathbf{T}_\tau(\mathbf{P}_1(F_n)) \cong I[0, \infty)$ for all n , so the source of $\phi_{1,\tau}$ is $I[0, \infty)$, while the target is 0. Thus $\phi_{1,\tau} : I[0, \infty) \rightarrow 0$ has nontrivial kernel and zero cokernel, hence $\mu_{1,\tau} = 1$, $u_{1,\tau} = 0$ (pure kernel).

Example A.11 (Mixed). Fix $\tau > 0$ and set

$$\mathbf{P}_1(F_n) = I[0, \tau - \frac{1}{n}] \oplus I[0, \infty),$$

with transition maps the obvious inclusions on the first summand and the identities on the second. Take F_∞ with $\mathbf{P}_1(F_\infty) = I[0, \infty) \oplus 0$, using cocone maps that send the second summand to 0. Then the first summand yields a cokernel contribution as in Example A.9, while the second yields a kernel contribution as in Example A.10. Hence $\mu_{1,\tau} = u_{1,\tau} = 1$ (mixed).

Remark A.12 (Realizability by filtered complexes). All persistence-level towers above are realizable by filtered complexes via the interval-realization assignment \mathcal{U} from Appendix B, up to f.q.i.; constructibility is preserved.

D.3. When $\phi_{i,\tau}$ is an isomorphism: $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$

Proposition A.13 (Isomorphism criterion). *Assume:*

- (i) *degreewise filtered colimits in $\text{FiltCh}(k)$ are computed objectwise on chains and filtrations;*
- (ii) *each $\mathbf{P}_i(F_n)$ lies in $\text{Pers}_k^{\text{ft}}$;*
- (iii) *\mathbf{T}_τ commutes with the filtered colimit of $\{\mathbf{P}_i(F_n)\}$ in $[\mathbb{R}, \text{Vect}_k]$, and the result is constructible (Appendix A);*
- (iv) *the cocone exhibits a colimit at persistence level: the canonical map $\text{colim}_n \mathbf{P}_i(F_n) \xrightarrow{\cong} \mathbf{P}_i(F_\infty)$ is an isomorphism in $[\mathbb{R}, \text{Vect}_k]$.*

Then for every i and $\tau > 0$, the comparison map $\phi_{i,\tau}$ is an isomorphism in $\text{Pers}_k^{\text{ft}}$. Consequently $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$.

Remark A.14. Condition (iv) is automatic if F_∞ is the colimit of $\{F_n\}$ in a filtered-complex model where \mathbf{P}_i is computed objectwise and the scope rule of Appendix A applies; no claim is made beyond that regime.

D.4. Sufficient conditions ensuring $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$

The summability condition $\sum_n d_{\text{int}}(\mathbf{P}_i(F_{n+1}), \mathbf{P}_i(F_n)) < \infty$ alone does not guarantee $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$; see §D.5. The following hypotheses are sufficient.

Theorem A.15 (Sufficient conditions for $\phi_{i,\tau}$ to be an isomorphism). *Fix i and $\tau > 0$. Each of the following implies that $\phi_{i,\tau}$ is an isomorphism (hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$):*

- (S1) **Commutation and apex colimit:** \mathbf{T}_τ commutes with the filtered colimit of $\{\mathbf{P}_i(F_n)\}$ in $[\mathbb{R}, \text{Vect}_k]$, the outcome is constructible, and the cocone exhibits a colimit at persistence level (Proposition A.13(iv)).
- (S2) **No τ -accumulation from below:** there exists $\eta > 0$ such that, for all sufficiently large n , no bar in $\mathbf{P}_i(F_n)$ has length in $(\tau - \eta, \tau)$. Equivalently, there is no sequence of bar lengths strictly increasing to τ .
- (S3) **\mathbf{T}_τ -Cauchy with compatible cocone:** the sequence $\mathbf{T}_\tau(\mathbf{P}_i(F_n))$ is Cauchy in the interleaving metric, and the cocone to $\mathbf{T}_\tau(\mathbf{P}_i(F_\infty))$ identifies the metric limit with the colimit target. (Here we use only completeness/uniqueness of limits for p.f.d. barcodes under bottleneck/interleaving distance.)

Proof. (S1) is Proposition A.13. For (S2), the gap prevents creation at the apex of new bars of length $> \tau$: every long bar in $\mathbf{T}_\tau(\mathbf{P}_i(F_\infty))$ must appear at some finite stage and stabilize, yielding bijectivity on interval factors. For (S3), completeness gives a unique metric limit; the stated compatibility identifies it with the colimit target, so $\phi_{i,\tau}$ is an isometry and hence an isomorphism in $\text{Pers}_k^{\text{ft}}$. \square

D.5. A counterexample: $\sum d_{\text{int}} < \infty$ yet $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \not\sqsubseteq (0, 0)$

Example A.16 (Summable increments, pure cokernel at the apex). Fix $\tau > 0$ and set $\ell_n = \tau - \sum_{m \geq n} 2^{-m} \uparrow \tau$, so that $\sum_n (\ell_{n+1} - \ell_n) = \sum_n 2^{-n} < \infty$. Let $M_n := I[0, \ell_n]$ with $M_n \hookrightarrow M_{n+1}$ the standard inclusions. Then

$$d_{\text{int}}(M_n, M_{n+1}) = \frac{1}{2}(\ell_{n+1} - \ell_n) = 2^{-(n+1)}, \quad \sum_n d_{\text{int}}(M_n, M_{n+1}) < \infty.$$

Let $\mathbf{P}_1(F_n) = M_n$, and choose an apex with $\mathbf{P}_1(F_\infty) = I[0, \infty)$. For every n , $\mathbf{T}_\tau(M_n) = 0$, while $\mathbf{T}_\tau(\mathbf{P}_1(F_\infty)) \cong I[0, \infty)$. Thus $\mu_{\text{tot}, \tau} = 0$, $u_{\text{tot}, \tau} = 1$ (pure cokernel), despite summable interleaving distances along the tower.

This shows that $\sum d_{\text{int}} < \infty$ alone is insufficient to force $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$.

D.6. Converse failures and the Type IV catalog

D.6.1. $\text{Ext}^1 = 0$ does not imply $\text{PH}_1 = 0$. Let $A \in D^{[-1, 0]}(k\text{-mod})$ with $H^{-1}(A) = 0$ and $H^0(A) \neq 0$, e.g. $A = V[0]$ for a nonzero k -space V . Then $\text{Ext}^1(A, k[0]) \cong \text{Hom}(H^{-1}(A), k) = 0$ by Appendix C (Lemma .104). Choose $F \in \text{FiltCh}(k)$ with $\mathbf{P}_1(F) \sqsubseteq 0$ (e.g. a single finite interval) and $\mathcal{R}(F) \simeq A$; this can be arranged up to f.q.i. using the interval-realization template (Appendix B). Hence $\text{Ext}^1(\mathcal{R}(F), k[0]) = 0$ while $\text{PH}_1(F) \not\sqsubseteq 0$, refuting the converse of the bridge globally.

D.6.2. Type IV (pure cokernel) at fixed τ . Example A.9 exhibits $\mu_{\text{tot}, \tau} = 0$, $u_{\text{tot}, \tau} > 0$ with $\mathbf{T}_\tau(\mathbf{P}_1(F_n)) = 0$ for all n but $\mathbf{T}_\tau(\mathbf{P}_1(F_\infty)) \not\sqsubseteq 0$.

D.6.3. Type IV (mixed). Example A.11 yields $\mu_{\text{tot}, \tau} > 0$ and $u_{\text{tot}, \tau} > 0$ simultaneously.

D.6.4. Realization notes. All persistence-level constructions above are realizable by filtered complexes via \mathcal{U} (Appendix B), up to f.q.i.; constructibility is preserved.

D.7. Restart/Summability for window pasting

All persistence-layer statements in this subsection are made *after* applying \mathbf{T}_{τ_k} on each window.

Definition A.17 (Per-window safety margin and pipeline budget). Let $\{W_k = [u_k, u_{k+1}]\}_{k \in K}$ be a MECE partition (Appendix A, Definition .71). On each window W_k , fix a collapse threshold $\tau_k > 0$. For a degree i , define the *pipeline budget*

$$\Sigma \delta_k(i) := \sum_{U \subseteq W_k} \left(\delta_U^{\text{alg}}(i, \tau_k) + \delta_U^{\text{disc}}(i, \tau_k) + \delta_U^{\text{meas}}(i, \tau_k) \right),$$

and the *safety margin* $\text{gap}_{\tau_k}(i) > 0$ as the configured slack for B-Gate⁺ on W_k and degree i . (Here $\sum_{U \subseteq W_k}$ ranges over the finite set of logged update-steps/units on W_k .)

Lemma A.18 (Restart inequality). Assume that, on window W_k , B-Gate⁺ passes with $\text{gap}_{\tau_k}(i) > \Sigma \delta_k(i)$, and that the transition to W_{k+1} is realized by a finite composition of deletion-type steps and ε -continuations (both measured after \mathbf{T}_{τ_k}). Then there exists $\kappa \in (0, 1]$, depending only on the admissible step class and the τ -adaptation policy, such that

$$\text{gap}_{\tau_{k+1}}(i) \geq \kappa (\text{gap}_{\tau_k}(i) - \Sigma \delta_k(i)).$$

Proof sketch. Deletion-type steps are non-increasing for the monitored indicators after \mathbf{T}_{τ_k} (Appendix E), and ε -continuations are 1-Lipschitz. Aggregating drifts yields the stated retention factor κ . \square

Definition A.19 (Summability). A run satisfies *Summability* (on a degree set $I \subset \mathbb{Z}$) if

$$\sum_{k \in K} \Sigma \delta_k(i) < \infty \quad (\forall i \in I).$$

A sufficient design is geometric decay of the windowed thresholds τ_k (hence of bins) and bounded per-window step counts.

Theorem A.20 (Pasting windowed certificates). *Let $\{W_k\}_k$ be MECE, and on each W_k let B-Gate⁺ pass with $\text{gap}_{\tau_k}(i) > \Sigma \delta_k(i)$ for all $i \in I$. If the Restart inequality (Lemma A.18) holds at every transition and Summability (Definition A.19) holds, then the concatenation of windowed certificates yields a global certificate on $\bigcup_k W_k$ for degrees $i \in I$.*

Convergence Manager ($\Sigma\delta$) — auditable pseudocode.

```
# Inputs:
# windows: list of MECE windows W_k
# degrees: monitored degree set I
# tau: list of collapse thresholds tau_k aligned with windows
# deltas: per-window lists of triples (delta_alg, delta_disc, delta_meas) per degree
# policy: either "geometric(r<1)" or "p_series(p>1)"
#
def convergence_manager(windows, degrees, tau, deltas, policy):
    total = {i: 0.0 for i in degrees}
    for k, Wk in enumerate(windows):
        for i in degrees:
            sigma = 0.0
            for (d_alg, d_disc, d_meas) in deltas[k][i]:
                sigma += d_alg + d_disc + d_meas # Quantale-additive
            assert gap_tau[k][i] > sigma, "Restart inequality failed"
            total[i] += sigma
        if policy.kind == "geometric":
            r = policy.r
            assert 0 < r < 1 and tau[k+1] <= r * tau[k]
        elif policy.kind == "p_series":
            p = policy.p
            assert p > 1
    return {i: (total[i] < float("inf")) for i in degrees}
```

D.8. Stability bands, τ -sweeps, and detection algorithm

Definition A.21 (Stability band via τ -sweep). Fix a window W and degree i . Let $\{\tau_\ell\}_{\ell=1}^L$ be an increasing τ -sweep. A contiguous block $\{\tau_a, \dots, \tau_b\}$ is a *stability band* if

$$\mu_{i, \tau_\ell} = u_{i, \tau_\ell} = 0 \quad \text{for all } \ell \in \{a, \dots, b\},$$

and the verdict persists upon *refining* the sweep (inserting new τ -values) without introducing μ or u in the band.

Proposition A.22 (Robust detection of stability bands). *Assume (S1)–(S3) of Theorem A.15 hold on W . Then any sufficiently fine τ -sweep admits stability bands covering all τ at which $\phi_{i, \tau}$ is an isomorphism; conversely, detecting a stability band by a sweep and its refinement certifies $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$ on the band.*

Proof sketch. Under (S1)–(S3), $\phi_{i,\tau}$ is an isomorphism on open neighborhoods of the corresponding τ -values. A fine sweep samples each neighborhood; refinement removes aliasing. The converse follows by Definition A.21. \square

Remark A.23 (Caveat: non-monotonicity in τ). There is no general monotonicity of $\mu_{i,\tau}$ or $u_{i,\tau}$ in τ . Stability bands may be separated by isolated τ -values where $\phi_{i,\tau}$ fails to be an isomorphism.

Detection algorithm (auditable). Given a sweep $\text{TauSweep} = \{\tau_\ell\}_{\ell=1}^L$:

- (A1) For each ℓ , compute ϕ_{i,τ_ℓ} and record $(\mu_{i,\tau_\ell}, u_{i,\tau_\ell})$.
- (A2) Extract maximal contiguous index ranges $[a, b]$ with $(\mu, u) = (0, 0)$.
- (A3) Refine by inserting midpoints $\tau' = \frac{1}{2}(\tau_\ell + \tau_{\ell+1})$ within each candidate band and recompute (μ, u) .
- (A4) Accept a band iff all refined points also yield $(0, 0)$.
- (A5) Emit a certificate with hashes of inputs, tower metadata, and flags indicating which of (S1)–(S3) were used.

D.9. Implementation guide: APIs, stubs, and tests

All persistence-layer computations are understood after applying \mathbf{T}_τ .

Lean stubs (illustrative).

```
namespace PH
structure Tower ( : Type) :=
  (F : →FiltCh)
  (apex : FiltCh)
  (toApex : ∀i : , ChainMap (F i) apex)

def P_i (i : ℤ) : FiltCh →Pers := -- assumed given
def T_tau (τ : ℝ) : Pers →Pers := -- exact, 1-Lipschitz

def phi (i : ℤ) (τ : ℝ) (T : Tower ℕ) : PersHom :=
  have src := colim (fun n => T_tau τ (P_i i (T.F n)))
  have tgt := T_tau τ (P_i i T.apex)
  comparison src tgt -- canonical

def gdim (M : Pers) : Nat := -- multiplicity of I[0,∞) in barcode

def mu (i : ℤ) (τ : ℝ) (T : Tower ℕ) : Nat := gdim (kernel (phi i τ T))
def nu (i : ℤ) (τ : ℝ) (T : Tower ℕ) : Nat := gdim (cokernel (phi i τ T))
end PH
```

Listing 1: Lean 4 stubs for towers and obstruction indices

Coq stubs (illustrative).

```

Module PH.
Record Tower := {
  F : nat -> FiltCh;
  apex : FiltCh;
  toApex : forall n, ChainMap (F n) apex
}.

Parameter P_i : Z -> FiltCh -> Pers.
Parameter T_tau : R -> Pers -> Pers. (* exact, 1-Lipschitz *)

Definition phi (i:Z) (tau:R) (T:Tower) : PersHom :=
  let src := colim (fun n => T_tau tau (P_i i (F T n))) in
  let tgt := T_tau tau (P_i i (apex T)) in
  comparison src tgt.

Parameter gdim : Pers -> nat.

Definition mu (i:Z) (tau:R) (T:Tower) : nat := gdim (kernel (phi i tau T)).
Definition nu (i:Z) (tau:R) (T:Tower) : nat := gdim (cokernel (phi i tau T)).
End PH.

```

Listing 2: Coq stubs for towers and obstruction indices

Sample tests.

- (T1) **T3 (Filtered-colim stability)**. Construct a tower whose apex is the filtered colimit and for which \mathbf{T}_τ commutes with colim. Verify $\phi_{i,\tau}$ is an iso and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$.
- (T2) **T7 (Toy towers)**. Instantiate the pure kernel, pure cokernel, and mixed towers of §D.2 and confirm $(\mu, u) = (1, 0), (0, 1), (1, 1)$, respectively.
- (T3) **T9 (No τ -accumulation)**. Create barcodes with an η -gap below τ ; confirm $\phi_{i,\tau}$ is iso.
- (T4) **T10 (Cauchy+compatibility)**. Build a Cauchy sequence in the bottleneck metric whose limit equals the apex post- \mathbf{T}_τ ; confirm iso.
- (T5) **T11 (Restart+Summability)**. Simulate windows and transitions satisfying Lemma A.18 and Definition A.19; verify global certificate via Theorem A.20.

D.10. Audit schema: run.yaml, JSON, and HDF5 layout

YAML fields (mandatory).

```

phi:
  idx:
    - i: 1
      tau: 0.75
      iso: true #  $\varphi_{\{i,\tau\}}$  isomorphism?
      mu: 0
      nu: 0
      flags:

```

```

    S1: true # used commutation+apex-colim
    S2: false
    S3: false
  iso_tail:
    passed: true # tail check on refined sweep
    refinement_levels: 2
  bands:
    - i: 1
      start_tau: 0.70
      end_tau: 0.82
      certified: true # ( $u$ )=(0,0) across band
      method: "sweep+refine"
  windows:
    collapse:
      tau_sweep: [0.5, 0.6, 0.7, 0.75, 0.8]
  persistence:
    phi_iso_tail: "strict"
  summability:
    policy: "geometric"
    r: 0.8
    total_delta:
      i=0: 0.137
      i=1: 0.092
  tower:
    edges:
      - src: 0; dst: 1; kind: "inclusion"
      - src: 1; dst: 2; kind: "inclusion"
  hash:
    inputs: "sha256:..."
    code: "sha256:..."

```

Listing 3: Minimal audit fields in run.yaml (bands included)

JSON snippet (optional).

```

{
  "phi": {
    "idx": [
      { "i": 1, "tau": 0.75, "iso": true, "mu": 0, "nu": 0,
        "flags": { "S1": true, "S2": false, "S3": false },
        "iso_tail": { "passed": true, "refinement_levels": 2 } }
    ],
    "bands": [
      { "i": 1, "start_tau": 0.70, "end_tau": 0.82,
        "certified": true, "method": "sweep+refine" }
    ]
  },
  "summability": { "policy": "geometric", "r": 0.8,
    "total_delta": { "i=0": 0.137, "i=1": 0.092 } }
}

```

HDF5 groups (canonical order; bands included). We store comparison data under `/phi` and tower meta-data under `/tower`. A minimal layout is:

Group/Dataset	Contents
<code>/phi/idx/i</code>	integer degrees i
<code>/phi/idx/tau</code>	real thresholds τ
<code>/phi/idx/iso</code>	boolean flags
<code>/phi/idx/mu</code>	nonnegative integers $\mu_{i,\tau}$
<code>/phi/idx/nu</code>	nonnegative integers $u_{i,\tau}$
<code>/phi/idx/flags</code>	bitmask/provenance for (S1,S2,S3)
<code>/phi/idx/iso_tail/passed</code>	boolean
<code>/phi/bands/i</code>	integer degrees i
<code>/phi/bands/start_tau</code>	real τ -starts
<code>/phi/bands/end_tau</code>	real τ -ends
<code>/phi/bands/certified</code>	boolean certification flags
<code>/phi/bands/method</code>	string provenance (e.g. sweep+refine)
<code>/tower/edges/src</code>	integer source indices
<code>/tower/edges/dst</code>	integer target indices
<code>/tower/edges/kind</code>	categorical: inclusion/deletion/epsilon

D.11. Additional formalities: τ -criticality and bandwise certification

Proposition A.24 (Piecewise constancy off critical thresholds). *Fix i and a tower. There exists a finite set $S \subset (0, \infty)$ consisting of relevant bar lengths and their finite sums/differences (within the monitored window/degree budget) such that $\mu_{i,\tau}$ and $u_{i,\tau}$ are locally constant on each connected component of $(0, \infty) \setminus S$.*

Proof sketch. In the constructible regime, changes in Ker/coker after \mathbf{T}_τ can occur only when τ crosses a length at which truncation toggles an interval from “deleted” to “kept” in a kernel/cokernel decomposition. Fixing a bounded set of degrees and a finite window budget yields a finite critical set. \square

Corollary A.25 (Bandwise certification). *If ϕ_{i,τ_0} is an isomorphism for some τ_0 lying in a connected component of $(0, \infty) \setminus S$, then it is an isomorphism on the entire component.*

D.12. Completion note and cross-module conventions

Remark A.26 (No further supplementation required). This appendix provides: (i) the definition and calculus of the tower obstruction indices (μ, u) (generic fiber dimensions after truncation), including V -metric invariance and window finiteness on definable covers; (ii) naturality of ϕ , subadditivity under composition, and additivity under finite direct sums; (iii) toy towers (pure kernel/cokernel/mixed) and a counterexample showing $\sum d_{\text{int}} < \infty$ does not force $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$; (iv) sufficient conditions (S1)–(S3) guaranteeing $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$; (v) a Restart/Summability framework (with auditable pseudocode) to paste windowed certificates into global ones; (vi) a robust τ -sweep procedure and stability bands to certify $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)(\tau)$ on contiguous τ -ranges; and (vii) implementation-grade audit schemas (YAML/JSON/HDF5), API stubs (Lean/Coq), and test items. All statements remain within the global guard-rails (constructible 1D persistence over a field; persistence-layer equalities after truncation; f.q.i. at the filtered-complex layer).

Cross-module conventions.

- Ext-tests (Appendix C) are always taken against $k[0]$: $\text{Ext}^1(\mathcal{R}((C_\tau F)|_W), k[0])$.
- Update monotonicity follows the global rule (Appendix E): *deletion-type* updates are non-increasing for windowed energies and spectral tails *after truncation*, whereas *inclusion-type* updates are stable (non-expansive).
- Type labels follow the global convention *Type I–II / Type III / Type IV*; tower defects $(\mu, u) \text{ eq}(0, 0)$ are *Type IV* at the declared scale(s) τ .

Appendix E. Spectral Indicators: Monotonicity, Stability, Counterexamples [Proof/Spec] (reinforced)

Standing conventions (Appendix E). Throughout this appendix:

- Matrices/operators are Hermitian unless stated otherwise. Eigenvalues are always listed in *ascending* order: $\lambda_1(H) \leq \lambda_2(H) \leq \dots \leq \lambda_n(H)$.
- The *upper-tail counting function* is

$$N_\theta(H) := \#\{j : \lambda_j(H) \geq \theta\} \quad (\theta \in \mathbb{R}),$$

with left/right limits $N_{\theta-0}(H), N_{\theta+0}(H)$ taken in θ .

- All spectral audits are *windowed* and performed *after* collapse on the B-side single layer. When we speak about operators arising from filtered complexes, we mean: form $C_\tau F$ (collapse), realize to a linear operator $L(C_\tau F)$ under the fixed realization policy (Appendix G), and *then* compute spectral indicators on each window.
- Equalities asserted about persistence objects are strict in $\text{Pers}_k^{\text{ft}}$; filtered–complex statements are *up to filtered quasi-isomorphism (f.q.i.)*. Filtered colimits follow the scope rule of Appendix A, Remark .67.
- Cross-module conventions (global): Ext-tests are taken against $k[0]$ (Appendix C), i.e. $\text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$; energy exponents are uniform $\alpha > 0$ (default $\alpha = 1$); type labels use *Type I–II / Type III / Type IV*.
- V -enrichment is governed by Appendix A, Remark .76: the V -metric/aggregation layer does not alter the underlying algebraic backbone.

For $\tau > 0$, define the *clipped spectrum*

$$\text{clip}_\tau(H) := (\min\{\lambda_j(H), \tau\})_{j=1}^n,$$

the *clipped sum*

$$S^{\leq \tau}(H) := \sum_{j=1}^n \min\{\lambda_j(H), \tau\},$$

and the *sub-threshold deficit*

$$D^{< \tau}(H) := \sum_{j=1}^n (\tau - \lambda_j(H))_+, \quad x_+ := \max\{x, 0\}.$$

We use the operator norm $\|\cdot\|_{\text{op}}$ and Frobenius norm $\|\cdot\|_{\text{fro}}$.

Deletion vs. inclusion (policy). When H arises by restricting admissible degrees of freedom, imposing Dirichlet constraints, eliminating internal dofs by shorting (Schur complement/Kron reduction), or taking principal submatrices, we call the step *deletion-type*. When H is obtained by adding degrees of freedom, couplings, or enlarging a domain, we call it *inclusion-type*. Deletion-type steps admit one-sided monotonicity (with a specified Loewner orientation); inclusion-type steps admit only stability (non-expansive) unless additional order hypotheses are imposed. *All inclusion-type claims are explicitly labeled as “non-expansive only.”*

E.0. Scope and window policy

All spectral audits in this appendix are *windowed* and performed *after* collapse on the B-side single layer. Comparisons follow the mandatory order:

$$\boxed{\text{for each } t \implies \text{apply } \mathbf{P}_t \implies \text{apply } \mathbf{T}_\tau \implies \text{compare in Pers}_k^{\text{ft}}}.$$

Spectral indicators are computed only as *auxiliary diagnostics* on $L(C_\tau F)$ under a fixed policy (Appendix G); they never replace B–Gate⁺.

When a spectral window $[a, b]$ with bin width $\beta > 0$ is used, bins are *half-open, right-attribution* $I_r = [a + r\beta, a + (r + 1)\beta)$, and eigenvalues at a right boundary are counted in the next bin. Underflows/overflows are recorded. This policy ensures reproducibility and compatibility with Overlap Gate and B–Gate⁺ (Appendix G; Chapter 1).

E.1. Deletion-type monotonicity (principal/Dirichlet, Schur complement, Loewner)

Proposition B.1 (Principal/Dirichlet restriction: interlacing and tail-count control). *Let $A \in \mathbb{R}^{n \times n}$ be Hermitian and let B be a principal $(n - 1) \times (n - 1)$ submatrix (obtained, e.g., by pinning a coordinate—Dirichlet restriction). Then Cauchy interlacing holds:*

$$\lambda_1(A) \leq \lambda_1(B) \leq \lambda_2(A) \leq \cdots \leq \lambda_{n-1}(B) \leq \lambda_n(A).$$

In particular, for every $\theta \in \mathbb{R}$,

$$N_\theta(B) \leq N_\theta(A), \quad N_\theta(B) \geq \max\{0, N_\theta(A) - 1\}.$$

Proposition B.2 (Schur complement (shorting) monotonicity). *Partition $M = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix} \succeq 0$ with $C \succ 0$ and form the Schur complement $S := A - BC^{-1}B^\top$. Then $S \preceq A$. Consequently, for all j and all $\theta \geq 0$,*

$$\lambda_j(S) \leq \lambda_j(A), \quad N_\theta(S) \leq N_\theta(A).$$

Proposition B.3 (Loewner-order monotonicity). *If $0 \preceq A \preceq B$ (Loewner order), then for each j , $\lambda_j(A) \leq \lambda_j(B)$, and for every $\theta \geq 0$, $N_\theta(A) \leq N_\theta(B)$.*

Remark B.4 (Heat traces and spectral tails). For PSD matrices and $t > 0$, the heat trace $\text{HT}(t; H) = \sum_j e^{-t\lambda_j(H)}$ satisfies: if $A' \preceq A$ (contraction), then $\text{HT}(t; A') \geq \text{HT}(t; A)$; if $A' \succeq A$ (hardening), then $\text{HT}(t; A') \leq \text{HT}(t; A)$. Likewise, for spectral tails $\text{ST}_\beta(H) = \sum_{j \geq 1} \lambda_j(H)^{-\beta}$ with $\beta > 0$, one has $\text{ST}_\beta(A') \geq \text{ST}_\beta(A)$ under $A' \preceq A$ and the reverse inequality under $A' \succeq A$, provided all $\lambda_j > 0$. In practice, tails are computed on $L(C_\tau F)$ with zero modes removed or handled by pseudoinverses; see Appendix G.

Corollary B.5 (Conservative averaging). *If $A_1, \dots, A_m \succeq 0$ satisfy $A_\ell \preceq A$ for all ℓ , then for any convex combination $\bar{A} := \sum_\ell w_\ell A_\ell$ with $w_\ell \geq 0$, $\sum_\ell w_\ell = 1$, $\bar{A} \preceq A$. Therefore $\lambda_j(\bar{A}) \leq \lambda_j(A)$, $N_\theta(\bar{A}) \leq N_\theta(A)$ for $\theta \geq 0$, and the heat trace/tail inequalities of Remark B.4 apply.*

Remark B.6 (Orientation for deletions). Two Loewner orientations occur in practice. *Contractions* (e.g. Schur complements, Kron reduction) produce $A' \preceq A$; *hardening* operations (e.g. some PDE Dirichlet comparisons across different media) may yield $A' \succeq A$. Deletion-type monotonicities in this appendix are always stated with the relevant orientation explicitly indicated.

E.2. Inclusion-type counterexamples

Deletion-type monotonicity does *not* extend naively to inclusion-type operations without additional order hypotheses.

Example B.7 (Neumann/domain inclusion reverses direction). For the Neumann Laplacian on an interval, enlarging the domain decreases the nonzero eigenvalues: on $[0, L]$, the first nonzero Neumann eigenvalue is $(\pi/L)^2$, so passing $L : 1 \rightarrow 2$ reduces it from π^2 to $(\pi/2)^2$. Thus any “inclusion \Rightarrow increase” heuristic fails under Neumann-type constraints.

Example B.8 (Indefinite coupling can move eigenvalues both ways). Let $A = I_2 = \text{diag}(1, 1)$ and $B = \begin{pmatrix} 1 & M \\ M & 1 \end{pmatrix}$ with $M > 1$. Then B has eigenvalues $1 - M$ and $1 + M$, so for $\theta = 0$, $N_\theta(B) = 1 < N_\theta(A) = 2$, while the top eigenvalue λ_2 increases. Without a Loewner relation ($B - A$ indefinite), no monotone law survives.

Example B.9 (Principal extension lacks a fixed direction). Let $B = [0]$ (eigenvalue 0) and $A = \begin{pmatrix} 0 & t \\ t & 0 \end{pmatrix}$ with $t \neq 0$. Going from B to A (adding one dof and a coupling) produces eigenvalues $-|t|$ and $|t|$: the maximum increases to $|t|$, but the minimum decreases to $-|t|$. Hence no uniform increase/decrease holds under inclusion.

These examples justify restricting monotone claims to the deletion/Loewner settings formalized in §E.1.

E.3. Continuity, stability, and truncated functionals (with V -metric lift)

Proposition B.10 (Weyl and Hoffman–Wielandt). For Hermitian $A, B \in \mathbb{R}^{n \times n}$,

$$\max_{1 \leq j \leq n} |\lambda_j(A) - \lambda_j(B)| \leq \|A - B\|_{\text{op}}, \quad \left(\sum_{j=1}^n |\lambda_j(A) - \lambda_j(B)|^2 \right)^{1/2} \leq \|A - B\|_{\text{fro}}.$$

Hence $A \mapsto (\lambda_1(A), \dots, \lambda_n(A))$ is 1-Lipschitz from $(\|\cdot\|_{\text{op}})$ into $(\mathbb{R}^n, \|\cdot\|_\infty)$.

Corollary B.11 (Lipschitz stability of clipped spectra). For any $\tau > 0$ and Hermitian A, B ,

$$\sum_{j=1}^n \left| \min\{\lambda_j(A), \tau\} - \min\{\lambda_j(B), \tau\} \right| \leq \sum_{j=1}^n |\lambda_j(A) - \lambda_j(B)| \leq \sqrt{n} \|A - B\|_{\text{fro}} \leq n \|A - B\|_{\text{op}}.$$

Consequently, $S^{\leq \tau}$ is \sqrt{n} -Lipschitz in $\|\cdot\|_{\text{fro}}$ and n -Lipschitz in $\|\cdot\|_{\text{op}}$.

Proposition B.12 (Semicontinuity of counting indicators). If $A_m \rightarrow A$ in operator norm and θ is not an eigenvalue of A , then $N_\theta(A_m) = N_\theta(A)$ for all large m (local constancy). In general,

$$\limsup_{m \rightarrow \infty} N_\theta(A_m) \leq N_{\theta-0}(A), \quad \liminf_{m \rightarrow \infty} N_\theta(A_m) \geq N_{\theta+0}(A).$$

Proposition B.13 (Truncated functionals: monotonicity and stability). Fix $\tau > 0$. For an $n \times n$ positive semidefinite (PSD) matrix A , set

$$S^{\leq \tau}(A) := \sum_{j=1}^n \min\{\lambda_j(A), \tau\}, \quad D^{< \tau}(A) := \sum_{j=1}^n (\tau - \lambda_j(A))_+.$$

Then:

(1) Deletion; Loewner contraction $A' \preceq A$. For all j , $\lambda_j(A') \leq \lambda_j(A)$, hence $N_\theta(A') \leq N_\theta(A)$ for every $\theta \geq 0$, and

$$S^{\leq \tau}(A') \leq S^{\leq \tau}(A), \quad D^{< \tau}(A') \geq D^{< \tau}(A).$$

(2) Deletion; Loewner hardening $A' \succeq A$. All inequalities in (1) reverse:

$$\lambda_j(A') \geq \lambda_j(A), \quad N_\theta(A') \geq N_\theta(A) \ (\theta \geq 0), \quad S^{\leq \tau}(A') \geq S^{\leq \tau}(A), \quad D^{< \tau}(A') \leq D^{< \tau}(A).$$

(3) Lipschitz stability. For any Hermitian A, B ,

$$|D^{< \tau}(A) - D^{< \tau}(B)| \leq \sum_{j=1}^n |\lambda_j(A) - \lambda_j(B)| \leq \sqrt{n} \|A - B\|_{\text{fro}} \leq n \|A - B\|_{\text{op}}.$$

Theorem B.14 (Deletion-type monotonicity and V -Lipschitz after collapse). *Let V be a commutative quantale and endow the per-window spectral dashboard (counts N_θ , clipped sums $S^{\leq \tau}$, deficits $D^{< \tau}$, heat traces, tails, cumulative profiles C_r) with a Lawvere V -metric aggregator (Appendix A, Remark .76). Then, measured after applying C_τ and evaluating under the fixed realization policy (Appendix G):*

- (a) *For every deletion-type update U , the dashboard is componentwise monotone in the appropriate Loewner orientation as in Propositions B.1–B.13 and Remark B.4.*
- (b) *For every ε -continuation with $\|A' - A\|_{\text{op}} \leq \varepsilon$, each scalar component changes by at most its explicit Lipschitz bound (Weyl/Hoffman–Wielandt and Corollary B.11), and any V -aggregation preserves the same (non-expansive) estimate.*

Spectral indicators are auxiliary: they never serve as sole gate criteria.

Remark B.15 (Heat traces and spectral tails: stability). Let $\text{HT}(t; H) = \sum_j e^{-t\lambda_j(H)}$, $\text{ST}_\beta(H) = \sum_j \lambda_j(H)^{-\beta}$ with zero modes removed. If $\|A - B\|_{\text{op}} \leq \varepsilon$, then on windows with $\lambda_j \geq \lambda_{\min}^+ > 0$, both $\text{HT}(t; \cdot)$ and $\text{ST}_\beta(\cdot)$ admit Lipschitz-type bounds in terms of $\sum_j |\lambda_j(A) - \lambda_j(B)|$, hence in terms of $\|\cdot\|_{\text{fro}}$ (and $\|\cdot\|_{\text{op}}$ with an n factor) by Proposition B.10. In practice, tails/heat traces are evaluated on $L(C_\tau F)$ under a fixed normalization (Appendix G).

E.3.5. Safe low-pass post-processing and 1-Lipschitz verification (final)

We record two safe low-pass mechanisms used as *post-collapse* auxiliaries. Both are *non-expansive* under the stated conditions and compatible with the dashboard above. They never replace B–Gate⁺.

(A) Temporal/index low-pass on a sequence $\{A_j\}_j$ of PSD operators. Fix a finitely supported kernel $h^{(\tau)} : \mathbb{Z} \rightarrow [0, 1]$ with:

$$\text{(LP1) evenness: } h^{(\tau)}[r] = h^{(\tau)}[-r],$$

$$\text{(LP2) unit mass: } \sum_{r \in \mathbb{Z}} h^{(\tau)}[r] = 1,$$

$$\text{(LP3) scale: } \text{supp}(h^{(\tau)}) \subseteq [-R(\tau), R(\tau)].$$

Define the smoothed sequence $\tilde{A}_j := \sum_r h^{(\tau)}[r] A_{j-r}$ (finite sum). Then:

(T1) *PSD and Loewner convexity.* $\tilde{A}_j \succeq 0$ and, if $A_{j-r} \preceq A^*$ for all r , then $\tilde{A}_j \preceq A^*$ (Corollary B.5).

(T2) *Non-expansive in $\|\cdot\|_{\text{op}}$.* For any sequences $\{A_j\}, \{B_j\}$,

$$\|\tilde{A}_j - \tilde{B}_j\|_{\text{op}} \leq \sum_r h^{(\tau)}[r] \|A_{j-r} - B_{j-r}\|_{\text{op}} \leq \sup_m \|A_m - B_m\|_{\text{op}}.$$

Thus the temporal low-pass is 1-Lipschitz in ℓ_∞ -in- j operator norm.

(T3) *Dashboard compatibility.* Apply the dashboard to each \tilde{A}_j after C_τ . Deletion-type monotonicity is preserved when the original updates are deletion-type and the kernel averages only within the same update class; otherwise only non-expansiveness is claimed.

(B) Spectral low-pass via functional calculus (filter scale σ , not the collapse threshold). Let $A \succeq 0$ and choose one of the *safe families* (parameter $\sigma > 0$):

- *Heat filter:* $f_\sigma(\lambda) := e^{-\sigma\lambda}$. Then

$$f_\sigma(A) - f_\sigma(B) = - \int_0^\sigma e^{-(\sigma-s)A} (A - B) e^{-sB} ds, \quad \Rightarrow \quad \|f_\sigma(A) - f_\sigma(B)\|_{\text{op}} \leq \sigma \|A - B\|_{\text{op}}.$$

Thus $\sigma \leq 1 \Rightarrow$ operator 1-Lipschitz.

- *Resolvent filter:* $r_\sigma(\lambda) := (1 + \lambda/\sigma)^{-1}$. By the resolvent identity,

$$r_\sigma(A) - r_\sigma(B) = (I + A/\sigma)^{-1} \frac{B - A}{\sigma} (I + B/\sigma)^{-1},$$

and $\|(I + A/\sigma)^{-1}\|_{\text{op}}, \|(I + B/\sigma)^{-1}\|_{\text{op}} \leq 1$. Hence

$$\|r_\sigma(A) - r_\sigma(B)\|_{\text{op}} \leq \frac{1}{\sigma} \|A - B\|_{\text{op}},$$

so $\sigma \geq 1 \Rightarrow$ operator 1-Lipschitz.

Both filters satisfy $0 \leq f_\sigma(\lambda), r_\sigma(\lambda) \leq 1$ and $f_\sigma(0) = r_\sigma(0) = 1$ (unit mass at DC), and they are operator-monotone decreasing in λ . When applied to $L(C_\tau F)$, the resulting operators keep the dashboard non-expansive; if a deletion update satisfies a Loewner orientation, the filtered indicators inherit the same orientation.

(C) 1-Lipschitz verification checklist (runtime, per window).

- (V1) *Temporal low-pass:* check (LP1)–(LP3) and $\sum h = 1$. Report `lp.kind="temporal"`, `lp.mass=1.0`, `lp.support`, and `lp.even=true`. Non-expansiveness holds with constant 1.
- (V2) *Heat filter:* record σ and ensure $\sigma \leq 1$ for 1-Lipschitz; otherwise `log lip_const= σ` and use it in the V -aggregator.
- (V3) *Resolvent filter:* ensure $\sigma \geq 1$ for 1-Lipschitz; otherwise `log lip_const= $1/\sigma$` .
- (V4) *Inclusion-type steps:* tag `inclusion="non-expansive-only"`; no monotone claim is emitted.

(D) Manifest snippet (auditable).

```
spectral_post:
  lowpass:
    mode: "temporal"          # or "heat" | "resolvent"
    kernel:
      even: true
      mass: 1.0
      support: [-2, 2]        # indices r with h[r] ≠ 0
      taps: [0.1, 0.2, 0.4, 0.2, 0.1]
    lipschitz:
      constant: 1.0
      verified: true
    inclusion_policy: "non-expansive-only"
# If mode is "heat":
#   param_sigma: 0.75
#   lipschitz.constant: 0.75
# If mode is "resolvent":
#   param_sigma: 2.0
#   lipschitz.constant: 0.5
```

Listing 4: run.yaml (spectral low-pass metadata)

E.4. Auxiliary spectral bars (aux-bars): definition, stability, and policy [Spec]

We formalize *auxiliary spectral bars* as diagnostics alongside persistence. They never replace B–Gate⁺ and are used only as auxiliary evidence.

E.4.1. Binning and endpoint convention. Fix a spectral window $[a, b]$ and a bin width $\beta > 0$. Let $R := \lfloor (b - a)/\beta \rfloor$. Define half-open, right-attribution bins $I_r = [a + r\beta, a + (r + 1)\beta)$ for $r = 0, 1, \dots, R - 1$. An eigenvalue at the bin’s right boundary is counted in the next bin. Record underflow $U(H) := \#\{j : \lambda_j(H) < a\}$ and overflow $O(H) := \#\{j : \lambda_j(H) \geq b\}$. For a Hermitian H , define the bin occupancy

$$E_r(H) := \#\{j : \lambda_j(H) \in I_r\},$$

and the cumulative (upper-tail) profile

$$C_r(H) := \sum_{s=r}^{R-1} E_s(H) = N_{a+r\beta}(H) - O(H).$$

E.4.2. Aux-bars across an index (time/tower). Let $(H_j)_{j \in J}$ be a sequence (time or tower). For fixed r , the set $\{j : E_r(H_j) > 0\}$ decomposes into maximal consecutive runs $J_{r,\ell}$. Each run $J_{r,\ell}$ defines an aux-bar $(r, J_{r,\ell})$ with lifetime $|J_{r,\ell}|$ (or a rescaled duration). We log aux-count = $\sum_{r,\ell} 1$, aux-mass = $\sum_r E_r(H_j)$ (per index j), and active bins = $\#\{r : E_r(H_j) > 0\}$.

E.4.3. Monotonicity/stability.

Proposition B.16 (Cumulative-profile monotonicity under deletion/Loewner). *If $A' \preceq A$ (PSD contraction) or A' is a principal/Dirichlet restriction of A , then for every r ,*

$$C_r(A') \leq C_r(A).$$

Proposition B.17 (Cumulative-profile stability). *If $\|A - B\|_{\text{op}} \leq \varepsilon$ and $q := \lceil \varepsilon/\beta \rceil$, then for all r ,*

$$C_{r+q}(B) \leq C_r(A) \leq C_{\max\{0, r-q\}}(B).$$

In particular, if $\varepsilon < \beta$, the cumulative profile can shift by at most one bin.

Corollary B.18 (Definable windows: finiteness and piecewise constancy). *On an o -minimal definable spectral window with finite Leray/Čech depth (Appendix H/J), the sequences $r \mapsto E_r(H_j)$ and $r \mapsto C_r(H_j)$ admit only finitely many bin-transition events per window and are piecewise constant in j . Aux-bars are therefore finite in number per window and auditable.*

Remark B.19 (Policy). • Deletion-type steps: enforce monotonicity on the *cumulative* profile C_r . Per-bin occupancies and lifetimes are diagnostics only.

- ε -continuations: with $\|A_{j+1} - A_j\|_{\text{op}} \leq \varepsilon$, declare stability up to $\pm q = \lceil \varepsilon/\beta \rceil$ bin shifts; record `eps_cont_shift_bins` in the manifest.
- Inclusion-type steps: claim no monotonicity; only stability bounds are used (*non-expansive only*).
- Under/overflow must be logged. Optional conservative rules (policy-only): $O = 0$, $C_{R-1} = 0$.

E.4.4. Reproducibility fields. The run manifest `run.yaml` should include (Appendix G):

- `spectral.range` $[a, b]$, `bin_width` β , bins R , endpoint policy half-open/right-attribution;
- underflow/overflow per index j ;
- `cum_profile`: the sequence $C_r(H_j)$ per j ;
- `auxBars` (optional): list of runs $(r, J_{r,\ell})$ with lifetimes;
- `eps_cont_bound`: ε and derived `eps_cont_shift_bins` $= \lceil \varepsilon/\beta \rceil$;
- `spectral_policy.order`: "ascending"; `spectral_policy.norm`: "op" or "fro"; bounds `lambda_min`, `lambda_max`, optional `lip_tol`.

E.5. Implementation and reproducibility: JSON/HDF5 schemas

JSON layout (mandatory fields).

```
{
  "meta": {
    "schema_version": "2025-03-15",
    "eigen_units": "dimensionless",
    "order": "ascending",
    "sorted": true,
    "norm": "op",           // or "fro"
    "Ntheta_convention": { "left": "N_{\theta-0}", "right": "N_{\theta+0}" },
    "window": { "range": [0.0, 2.0], "semantics": "closed" },
    "clip_tau": 1.0,
    "tol_eig": 1e-8,
    "aux_policy": { "bin": 0.02, "right_attribution": true },
    "coverage_check": { "thetas_in_window": true },
    "links": { "run_id": "...", "run_yaml_hash": "sha256:..." }
  }
}
```

```

    },
    "operators": [
      {
        "id": "sha256:...A",
        "kind": "laplacian_dirichlet",
        "n": 500,
        "spectrum": { "eigs": [0.10, 0.12, 0.45, ...] }, // ascending
        "clip": { "tau": 1.0, "sum": 37.219, "deficit": 12.004 },
        "underflow": 0, "overflow": 0
      },
      {
        "id": "sha256:...B",
        "kind": "principal_submatrix",
        "parent": "sha256:...A",
        "N_theta": [
          { "theta": 0.20, "left": 17, "right": 16 },
          { "theta": 0.50, "left": 10, "right": 10 }
        ],
        "cum_profile": [ 13, 11, 8, 2, 0 ],
        "underflow": 2, "overflow": 0,
        "monotonicity": { "type": "deletion", "passed": true }
      }
    ],
    "lowpass": {
      "mode": "heat", // "temporal" | "resolvent"
      "param_sigma": 0.75,
      "lipschitz_constant": 0.75,
      "verified": true,
      "inclusion_policy": "non-expansive-only"
    },
    "hash": "sha256:...spec"
  }

```

Listing 5: Minimal spec.json layout (one run, multiple operators)

HDF5 layout (canonical).

- Datasets: /spec/ops/{id}/eig (float64 ascending), /spec/ops/{id}/clip/sum (float64), /spec/ops/{id}/clip/deficit (float64), /spec/ops/{id}/Ntheta/theta,left,right (parallel arrays), /spec/ops/{id}/cum_profile (int32), /spec/ops/{id}/underflow (int32), /spec/ops/{id}/overflow (int32), /spec/lowpass/mode (fixed string), /spec/lowpass/param_sigma (float64), /spec/lowpass/lipschitz_constant (float64), /spec/lowpass/verified (bool).
- Attributes: order="ascending", norm="op"|"fro", eigen_units, tol_eig, schema_version, bin policy, and canonical HDF5 flags (track_times=false, UTF-8 fixed strings, fixed chunking); see Appendix G.

E.6. Tests and operational checklist

Core tests.

1. **T2 (Deletion-type monotonicity).** For principal/Dirichlet or Schur complements, verify N_θ , $S^{\leq \tau}$, $D^{< \tau}$, heat trace, and tails satisfy the monotone direction consistent with the Loewner orientation. Log `monotonicity.passed=true`.
2. **T1 (ε -continuation stability).** Given $\|A_{j+1} - A_j\|_{\text{op}} \leq \varepsilon$, validate the bin-shift bounds in Proposition B.17 and Lipschitz bounds for $S^{\leq \tau}$, $D^{< \tau}$.
3. **T9 (Coverage).** Confirm that all θ -queries, bin windows, and eigenvalues fall within declared windows; if not, log underflow/overflow and set `coverage_check.*=false` with a justification.
4. **T12 (Low-pass safety).** For temporal low-pass, check (LP1)–(LP3) and $\sum h = 1$; for heat/resolvent, assert the scale choices yield operator Lipschitz constant ≤ 1 (or record the constant if > 1) and verify against $\|A - B\|_{\text{op}}$.

Operational checklist (per window).

- Fix spectral policy: `order="ascending"`, norm selection, bin width β , spectral window $[a, b]$.
- Compute spectra on $L(C_\tau F)$ and clip at the same τ used by persistence.
- Optionally apply a *safe* low-pass (temporal, heat, resolvent) with recorded scale σ and verified Lipschitz constant.
- Log underflow/overflow, cumulative profiles C_r , optional aux-bars with lifetimes (diagnostic).
- For deletion-type steps, assert C_r monotonicity; for ε -continuations, assert bin-shift stability with $\lceil \varepsilon/\beta \rceil$; for inclusion-type steps, *non-expansive only*.

E.7. Completion note

Remark B.20 (No further supplementation required). This appendix provides a complete, IMRN/AiM-ready treatment of spectral indicators consistent with the v16.0 guard-rails: (i) deletion-type monotonicities (principal/Dirichlet, Schur, Loewner) for N_θ , $S^{\leq \tau}$, $D^{< \tau}$, heat traces, and tails; (ii) inclusion-type counterexamples; (iii) Lipschitz stability via Weyl/Hoffman–Wielandt and induced bounds for clipped sums/deficits, heat traces, and tails; (iv) a windowed, half-open binning policy with cumulative-profile monotonicity and stability under ε -continuations; (v) V -metric reinforcement (Theorem B.14) ensuring deletion-type monotonicity and V -stable aggregation after collapse; (vi) **safe low-pass post-processing** with even-kernel/unit-mass/ τ -scale (temporal) and operator-Lipschitz filters (heat/resolvent) together with an auditable Lipschitz verification checklist; and (vii) reproducibility and canonical schemas (JSON/HDF5) with a minimal test suite (T1/T2/T9/T12). All claims are made after collapse on the B-side single layer, per-window, and integrate with δ -ledger accounting, Overlap Gate, and B–Gate⁺ elsewhere in the manuscript. No further supplementation is required for operational deployment or audit.

Appendix F. Formalization Sketch (Lean/Coq) [Spec] (reinforced)

Declaration C.1 (Lean/Coq stubs). Expose minimal, reusable interfaces: `pers_Ttau_exact`, `pers_Ttau_lipschitz`, `Ctau_lift`, `Ctau_colim`, `Ctau_pullback`, `mu_nu_vanish`, `PH1_to_Ext1_under_B`, `delta_pipeline_additivity`, enriched metrics `V_metric_shift`, and the canonical normal form `cnf_after_Ttau` (interval/Smith-type decomposition for kernels/cokernels). Equalities are confined to the persistence layer; filtered-level facts are packaged “up to f.q.i.”. AWFS stubs, *o-minimal* Čech, and Iwasawa-style control catalysts are provided as portable axioms with clear call sites.

Purpose (Appendix F). This appendix provides a fully integrated, implementation-oriented *Spec* for mechanizing the core claims of Appendices A–E in Lean/Coq with a module decomposition tailored for a minimal, portable “mini-library.” The categorical spine consists of the Serre localization and the reflector \mathbf{T}_τ (exact, idempotent), its 1-Lipschitz property on barcodes (interleaving metric), tower diagnostics (μ, u) via the comparison map $\phi_{i,\tau}$, and the one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ under the amplitude ≤ 1 realization policy. We work in the constructible (p.f.d.) range and adhere to the filtered-colimit scope rule (Appendix A, Remark .67). Cross-module conventions: Ext-tests are always against $k[0]$ (Appendix C), i.e. $\text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$; the energy exponent is globally $\alpha > 0$ (default $\alpha = 1$); type labels use *Type I–II / Type III / Type IV* for tower diagnostics. Spectral monotonicity is invoked only for deletion-type operations; inclusion-type operations are used solely with stability bounds (Appendix E).

Refereeing style (IMRN/AiM). Statements are modular with explicit hypotheses and reusable APIs. All constructions remain within abelian categories, exact localizations, and derived categories with bounded t -structures. Proof obligations used in the code stubs are isolated and cited to Appendices A–E; replacing admit/Axiom by library lemmas yields a fully checked artifact.

F.0. Reading guide and module map

We split the development into five modules and three thin catalysts:

- **AK.Core** (F.1–F.6, F.11): $\text{Pers}_k^{\text{ft}}$, the Serre subcategory \mathcal{E}_τ , the reflector \mathbf{T}_τ (exact, idempotent), interleaving/shift calculus and 1-Lipschitz, the collapse on filtered complexes C_τ , canonical normal form (CNF) after \mathbf{T}_τ , and invariance under filtered quasi-isomorphisms.
- **AK.LocalEquiv** (F.7): Window-local bridge $\text{PH} \leftrightarrow \text{Ext}$ under amplitude ≤ 1 , saturation/no-accumulation triggers, and tail identification for $\phi_{i,\tau}$.
- **AK.Tower** (F.8): Comparison map $\phi_{i,\tau}$, diagnostics (μ, u) , functoriality, stability bands, direct sums, compositions, and cofinal invariance.
- **AK.Gluing** (F.9, F.18): Overlap Gate (collapse-compatibility, A/B checks, Čech– Ext^1 , stability bands) and MECE windowing glue to a global verdict, including Restart/Summability via a DP-style manager.
- **AK.Spectral** (F.10): Spectral auxiliaries (Appendix E) used after collapse, with deletion-type monotonicity only; inclusion-type steps are *non-expansive only*.
- **Catalysts (thin)**: V -enriched metric/shift (F.A), AWFS skeleton (F.B) for functorial factorization used by C_τ , and *o-minimal* Čech + Iwasawa-style control (F.C) to streamline Overlap Gate.

Test fixtures include T7 (saturation gate), T10 (A/B overlap), T13 (δ -budget and restart chain).

F.1. Environment and objects [Spec] (AK.Core)

Fix a field k . Let Vect_k be the abelian category of finite-dimensional k -vector spaces and $[\mathbb{R}, \text{Vect}_k]$ the functor category (index (\mathbb{R}, \leq)). Let $\text{Pers}_k^{\text{ft}} \subset [\mathbb{R}, \text{Vect}_k]$ be the full subcategory of constructible persistence modules.

Let $\text{FiltCh}(k)$ be filtered chain complexes of finite-dimensional k -spaces, bounded in homological degree, with filtration-preserving maps. For $i \in \mathbb{Z}$ write $\mathbf{P}_i : \text{FiltCh}(k) \rightarrow \text{Pers}_k^{\text{ft}}$ for the degree- i persistence functor. The bar-deletion reflector $\mathbf{T}_\tau : \text{Pers}_k^{\text{ft}} \rightarrow \text{Pers}_k^{\text{ft}}$ is recalled from Appendix A.

Remark C.2 (Generic fiber dimension and stabilization). We adopt Appendix D, Remark A.2. For $M \in \text{Pers}_k^{\text{ft}}$, the generic fiber dimension is the multiplicity of the infinite interval $I[0, \infty)$ in the barcode of M ; equivalently,

$$\text{gdim}(M) = \lim_{t \rightarrow +\infty} \dim_k M(t),$$

which stabilizes in the constructible range. After applying \mathbf{T}_τ , kernels and cokernels again lie in $\text{Pers}_k^{\text{ft}}$, and gdim is computed there.

Specification C.3 (Stabilization lemma for constructible modules). If $M \in \text{Pers}_k^{\text{ft}}$, then there exist $T_0 \in \mathbb{R}$ and $c \in \mathbb{N}$ such that $\dim_k M(t) = c = \text{gdim}(M)$ for all $t \geq T_0$. Use: define $\text{gdim}(M)$ by this stabilized value c and prove iso-invariance of gdim .

Remark C.4 (Tower verdict normal form). We fix the diagnostic normal form:

$$\text{DiagZero} \quad :\Longleftrightarrow \quad (\mu, u) = (0, 0),$$

with μ, u defined from $\phi_{i,\tau}$ in §F.8. Type IV corresponds to $(\mu, u) \text{ eq}(0, 0)$ (Appendix D).

F.2. Serre subcategory and localization [Spec] (AK.Core)

Let $E_\tau \subset \text{Pers}_k^{\text{ft}}$ be the Serre subcategory generated by intervals of length $\leq \tau$. By Appendix A, E_τ is hereditary Serre and the inclusion $\iota_\tau : E_\tau^\perp \hookrightarrow \text{Pers}_k^{\text{ft}}$ admits an exact left adjoint $\mathbf{T}_\tau : \text{Pers}_k^{\text{ft}} \rightarrow E_\tau^\perp$ (reflector), inducing an equivalence

$$\text{Pers}_k^{\text{ft}}/E_\tau \simeq E_\tau^\perp.$$

Basic laws (API):

$$\mathbf{T}_\tau \circ \mathbf{T}_\tau \cong \mathbf{T}_\tau, \quad \mathbf{T}_\tau \dashv \iota_\tau.$$

Moreover, \mathbf{T}_τ is exact and preserves finite (co)limits (Appendix A).

F.3. Interleavings, shifts, and 1-Lipschitz [Spec] (AK.Core)

The interleaving pseudometric d_{int} on $\text{Pers}_k^{\text{ft}}$ is implemented via shift functors Shift_ε and ε -interleavings. Appendix A yields natural isomorphisms $\text{Shift}_\varepsilon \circ \mathbf{T}_\tau \simeq \mathbf{T}_\tau \circ \text{Shift}_\varepsilon$ that transport interleavings, hence

$$d_{\text{int}}(\mathbf{T}_\tau M, \mathbf{T}_\tau N) \leq d_{\text{int}}(M, N).$$

A V -enriched (Lawvere) lift is provided in §F.A.

F.4. Canonical Normal Form (CNF) after \mathbf{T}_τ [Spec] (AK.Core)

Definition C.5 (CNF for morphisms after truncation). For $f : M \rightarrow N$ in $\text{Pers}_k^{\text{ft}}$, define its canonical normal form $\text{CNF}_\tau(f)$ as the barcode-level decomposition of $\mathbf{T}_\tau f$ (Appendix A):

$$\mathbf{T}_\tau M \cong \bigoplus_a I_a, \quad \mathbf{T}_\tau N \cong \bigoplus_b I_b,$$

and, under these decompositions,

$$\mathbf{T}_\tau f \rightsquigarrow \left(\bigoplus_c \text{id}_{I_c} \right) \oplus \left(\bigoplus_d 0_{I_d} \right) \oplus \left(\bigoplus_e \iota_e \right),$$

where each summand is either an isomorphism on an interval, the zero map, or a standard inclusion $\iota_e : I[\ell, \infty) \rightarrow I[\ell', \infty)$ with $\ell \geq \ell'$. This CNF is unique up to permutation of factors.

Proposition C.6 (Reading μ, u from CNF). *Let f be as above. In $\text{CNF}_\tau(f)$, the multiplicity of $I[0, \infty)$ in $\text{Ker}(\mathbf{T}_\tau f)$ (resp. $\text{coker}(\mathbf{T}_\tau f)$) equals the tower invariant μ (resp. u) defined in §F.8. Finite bars contribute zero to μ, u .*

Remark C.7 (CNF as a proof/automation device). The CNF provides a canonical target for normalization tactics (§F.19). It is functorial under isomorphism and stable under cofinal tower reindexings.

F.5. Filtered colimits, scope rule, and constructibility [Spec] (AK.Core)

Scope rule. All filtered (co)limit computations are performed objectwise in $[\mathbb{R}, \text{Vect}_k]$, where filtered colimits are exact; they are invoked only under Appendix A, Remark .67. Whenever the result might exit $\text{Pers}_k^{\text{ft}}$, we either (i) verify constructibility, or (ii) compute outside and *return* via \mathbf{T}_τ (or an explicit finite-type truncation). No claim is made outside this regime.

F.6. Collapse on filtered complexes and f.q.i. [Spec] (AK.Core)

We use a collapse/threshold operation C_τ at the level of filtered complexes:

$$C_\tau : \text{FiltCh}(k) \longrightarrow \text{FiltCh}(k),$$

compatible with \mathbf{P}_i by construction, and preserving filtered quasi-isomorphisms (f.q.i.) up to localization. In practice, $\mathbf{P}_i \circ C_\tau$ is compared to $\mathbf{T}_\tau \circ \mathbf{P}_i$ via a natural transformation that becomes an isomorphism after applying \mathbf{T}_τ (Appendix A). All Ext-tests are taken after $\mathcal{R}(C_\tau F)$ with \mathcal{R} of amplitude $[-1, 0]$ (Appendix C). AWFS scaffolding for C_τ is given in §F.B.

F.7. Local equivalences within a window [Spec] (AK.LocalEquiv)

We formalize the window-local bridge $\text{PH} \leftrightarrow \text{Ext}$ under amplitude ≤ 1 and mild regularity.

Let a right-open window W be fixed; assume: (i) the filtered complex F is concentrated in homological degrees ≤ 1 on W , (ii) the collapse C_τ is stable on W (saturation), (iii) no τ -accumulation of critical values on W (Appendix D), (iv) tail identification for $\phi_{i,\tau}$ on W (Appendix D, (S1)).

Let \mathcal{R} be a derived realization of amplitude $[-1, 0]$ (Appendix C). Then there are natural identifications

$$H^{-1}(\mathcal{R}(C_\tau F)) \simeq \varinjlim_{t \in W} H_1(F^t), \quad \text{Ext}^1(\mathcal{R}(C_\tau F), k) \simeq \text{Hom}(H^{-1}(\mathcal{R}(C_\tau F)), k).$$

Thus, window-locally,

$$\text{PH}_1(F|_W) = 0 \implies \text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0.$$

Remark C.8 (Local reverse under $E_1(W) = 0$). Under the definable-window trigger $E_1(W) = 0$ and finite Leray depth (Appendix C, Corollary .112), $\text{Ext}^1(\mathcal{R}(C_\tau F|_W), k) = 0$ also implies $\text{PH}_1(C_\tau F|_W) = 0$. We encapsulate this as the tactic *reverse_bridge!* in §F.19.

F.8. Towers, $\phi_{i,\tau}$, and diagnostics (μ, u) [Spec] (AK.Tower)

We define towers, the comparison map $\phi_{i,\tau}$, and the invariants

$$\mu_{i,\tau}(T) := \text{gdim ker } \phi_{i,\tau}(T), \quad u_{i,\tau}(T) := \text{gdim coker } \phi_{i,\tau}(T),$$

computed in $\text{Pers}_k^{\text{ft}}$ after applying \mathbf{T}_τ (Appendix D). They are invariant under filtered quasi-isomorphisms of towers and under cofinal reindexings, and vanish under (S1)–(S3) (Appendix D). CNF (§F.4) provides a canonical device to read (μ, u) . Type labels are assigned by (μ, u) per Appendix D and Remark C.4.

F.9. Overlap Gate, MECE windows, and gluing [Spec] (AK.Gluing)

We formalize the operational glue from windows to a global verdict. MECE coverage, Čech–Ext¹ consistency on overlaps, Restart/Summability, and stability bands follow Appendices C–E. All overlap checks are carried out *after* collapse on the B-side single layer.

F.10. Spectral calculus and Lipschitz bounds [Spec] (AK.Spectral)

Spectral indicators are auxiliary and computed only after collapse, under the fixed window policy (Appendix E). Deletion-type spectral monotonicity is used only in the Loewner-orientable cases (principal/Dirichlet, Schur complement, Loewner order); inclusion-type operations are *non-expansive only*. Safe low-pass post-processing (temporal kernels; heat/resolvent filters with filter-scale parameter σ) is non-expansive under the recorded Lipschitz constant (Appendix E).

F.11. Edge identification $\text{PH}_1 \Rightarrow \text{Ext}^1$ [Spec] (AK.Core)

Let $\mathcal{R} : \text{FiltCh}(k) \rightarrow D(\text{Vect}_k)$ be of amplitude $[-1, 0]$. There is a natural edge identification

$$H^{-1}(\mathcal{R}(F)) \cong \varinjlim_t H_1(F^t).$$

For $A \in D^{[-1, 0]}$ we have $\text{Ext}^1(A, k) \cong \text{Hom}(H^{-1}(A), k)$. Combining these we obtain, for any F ,

$$\text{PH}_1(F) = 0 \implies \text{Ext}^1(\mathcal{R}(F), k) = 0,$$

and the same implication after insertion of the collapse C_τ . On definable right-open windows with $E_1(W) = 0$, the converse holds after collapse (Appendix C).

F.12. Lean 4 sketch (representative stubs) [Spec]

```
-- AK.Core: categories, Serre reflector, interleavings, scope rule, collapse, CNF
namespace AK.Core
open scoped BigOperators Classical
noncomputable section

variable (k : Type*) [Field k]
abbrev Vect := FinVect k
structure RIdx := ( : Type) (str : Preorder )
abbrev Diag := (RIdx → Vect k)

abbrev Pers := { M : Diag k // Constructible M }

-- Serre reflector T_τ : Pers → E_τ⊥
def Eτ (τ : ℝ≥0) : SerreSubcategory (Pers k) := by admit
noncomputable def Tτ (τ : ℝ≥0) : Pers k → (Eτ k τ).orthogonal := by admit
noncomputable def iotaτ (τ) : (Eτ k τ).orthogonal → Pers k := by admit
theorem Tτ_exact (τ) : (Tτ k τ).IsExact := by admit
theorem Tτ_idem (τ) : (Tτ k τ) ≫ (Tτ k τ) ≅ (Tτ k τ) := by admit
theorem Tτ_adj (τ) : (Tτ k τ) ⊣ (iotaτ k τ) := by admit
```

```

-- Interleaving stability (1-Lipschitz)
class Interleaving (C : Type*) :=
  (dist : C → C → ℝ≥0∞) (isPseudoMetric : PseudoMetricSpace C)
def d_int := (Interleaving.dist : Pers k → Pers k → ℝ≥0∞)
noncomputable def Shift (ε : ℝ≥0) : Pers k → Pers k := by admit
axiom shift_comm (τ ε) : Shift k ε >>> (Tτ k τ) ≅ (Tτ k τ) >>> Shift k ε
theorem Tτ_non-expansive (τ) :
  ∀ M N : Pers k, d_int k ((Tτ k τ).obj M) ((Tτ k τ).obj N) ≤ d_int k M N := by admit

-- Scope rule hooks
theorem filtered_colim_exact :
  ∀ {J} [IsFiltered J] (F : J → Vect), ExactFilteredColim F := by admit
axiom return_to_constructible :
  ∀ (D : SomeFilteredDiagram), Constructible (colim D)

-- Filtered complexes, persistence, collapse
abbrev FiltCh := FiltChCat k
def P_i (i : ℤ) : FiltCh k → Pers k := by admit
noncomputable def Cτ (τ : ℝ≥0) : FiltCh k → FiltCh k := by admit
theorem Cτ_preserves_fqi (τ) : PreservesFQI (Cτ k τ) := by admit

-- CNF after T_τ (barcode-normal form)
structure CNF where
  iso_parts : List Interval
  zero_parts : List Interval
  incl_parts : List (Interval × Interval) -- standard inclusions
noncomputable def cnf_after_Tτ {M N : Pers k} (τ) (f : M → N) : CNF := by admit

end AK.Core

```

F.13. Coq sketches (mathcomp/coq-category-theory) [Spec]

From mathcomp Require Import all_ssreflect all_algebra.
 From CoqCT Require Import Category Abelian Functor Limits Colimits.
 Set Implicit Arguments. Unset Strict Implicit. Unset Printing Implicit Defensive.

Module AK.

```

Parameter k : fieldType.
Axiom Vect : AbelianCat. (* f.d. k-vector spaces *)
Axiom Rposet : PreOrder. (* (ℝ, ≤), schematic *)
Definition Diag := FunctorCat Rposet Vect.
Parameter Constructible : Diag -> Prop.
Record Pers := { M : Diag; pfd : Constructible M }.

```

```

Axiom Eτ : SerreSubcat Pers.
Axiom Tτ : Functor Pers (Orthogonal Eτ).
Axiom iotaτ : Functor (Orthogonal Eτ) Pers.

```

Axiom $T\tau_exact : \text{ExactFunctor } T\tau.$
 Axiom $T\tau_idem : \text{FunctorComp } T\tau \ T\tau \cong T\tau.$
 Axiom $T\tau_adj : \text{Adjunction } T\tau \ \text{iota}\tau.$

Parameter $dint : \text{Pers} \rightarrow \text{Pers} \rightarrow \mathbb{R}.$
 Parameter $\text{Shift} : \mathbb{R} \rightarrow \text{Functor Pers Pers}.$
 Axiom $\text{shift_comm} : \text{forall } \text{eps}, \text{FunctorComp } (\text{Shift eps}) \ T\tau \cong \text{FunctorComp } T\tau \ (\text{Shift eps}).$
 Axiom $T\tau_non\text{-expansive} :$
 $\text{forall } (X \ Y : \text{Pers}), \text{dint } (T\tau \ X) \ (T\tau \ Y) \leq \text{dint } X \ Y.$

(* Collapse and persistence *)
 Parameter $\text{FiltCh} : \text{Type}.$
 Parameter $P_i : \mathbb{Z} \rightarrow \text{Functor FiltCh Pers}.$
 Parameter $C\tau : \mathbb{R} \rightarrow \text{Functor FiltCh FiltCh}.$
 Axiom $C\tau_preserves_fqi : \text{forall } \tau, \text{PreservesFQI } (C\tau \ \tau).$

(* CNF after $T\tau$ *)
 Record $\text{CNF} := \{ \text{iso_parts} : \text{seq Interval}; \text{zero_parts} : \text{seq Interval};$
 $\text{incl_parts} : \text{seq (Interval * Interval)} \}.$
 Parameter $\text{cnf_after_}T\tau : \text{forall } M \ N \ (f : \text{Hom } M \ N) \ \tau, \text{CNF}.$

End AK.

F.14. Tests and fixtures (T7, T10, T13) [Spec]

T7 (Saturation gate). Construct a tower with pure cokernel defect and one with a stationary summand, and their direct sum. Verify that: (i) μ, u equal the multiplicity of $I[0, \infty)$ after $T\tau$ (via CNF); (ii) cofinal reindexing $n \mapsto n+1$ preserves (μ, u) ; (iii) under (S1) the comparison $\phi_{i,\tau}$ is an isomorphism and DiagZero .

T10 (A/B overlap). Instantiate an η -tolerant A/B test on overlaps and verify soft commuting; check that window-local Ext^1 -vanishing glues via Čech- Ext^1 (Appendix C).

T13 (δ -budget). Generate a δ -ledger per window/degree; check additivity/post-stability and the restart inequality

$$\text{gap}_{k+1} \geq \kappa(\text{gap}_k - \Sigma \delta_k)_+.$$

Verify summability $\sum_k \Sigma \delta_k < \infty$ and that B-Gate^+ accepts precisely when $\text{gap} > \text{dsum}$ along with $\text{PH}_1 = 0$, $\text{Ext}^1 = 0$, and DiagZero .

F.15. What is proved, what is assumed [Spec]

- (*Localization*) E_τ is hereditary Serre; $T\tau$ exists, is exact, idempotent, and induces $\text{Pers}_k^{\text{ft}}/E_\tau \simeq E_\tau^\perp$.
- (*Stability*) $T\tau$ is 1-Lipschitz for the interleaving metric via shift-commutation (Appendix A).
- (*Towers*) $\phi_{i,\tau}$ is functorial; under (S1)–(S3) (Appendix D) it is an isomorphism, hence DiagZero . (μ, u) are invariant under f.q.i. and cofinal reindexings; finiteness holds by degree bounds.

- (*Bridge*) For $F \in \text{FiltCh}(k)$, \mathcal{R} has amplitude $[-1, 0]$; $H^{-1}(\mathcal{R}(F)) \simeq \varinjlim_t H_1(F^t)$ and $\text{Ext}^1(A, k) \simeq \text{Hom}(H^{-1}(A), k)$ for $A \in D^{[-1, 0]}$, hence $\text{PH}_1(F) = 0 \Rightarrow \text{Ext}^1(\mathcal{R}(F), k) = 0$. On definable right-open windows with $E_1(W) = 0$, the local reverse holds after C_τ (Appendix C).
- (*Spectral*) Deletion-type operators are spectrally monotone in the appropriate Loewner orientation; inclusion-type operators are controlled via stability bounds; safe low-pass is non-expansive under its recorded Lipschitz constant (Appendix E).

F.16. Notes on libraries and portability [Spec]

The Lean sketch targets mathlib (abelian categories, Serre subcategories, localization, derived categories). The Coq sketch targets mathcomp+coq-category-theory (or UniMath). Nontrivial steps are isolated behind admit/Axiom with explicit references to Appendices A–E. Replacing them by library lemmas yields a complete development. In Lean, define $\phi_{i,\tau}$ via Limits.colimit.desc and use colimit.hom_ext for naturality; in Coq, use Colim.desc/colim_map with right-to-left compose convention.

F.17. Thin catalysts (V-enrichment, AWFS, o-minimal Čech, Iwasawa control)

F.A. V -enriched metric/shift (Spec). Let V be a commutative quantale. Equip $\text{Pers}_k^{\text{ft}}$ with a Lawvere V -metric d_V obtained from d_{int} via a monotone embedding and a V -aggregator. Require: (i) d_V extends d_{int} on scalars, (ii) Shift_ε is V -1-Lipschitz, (iii) \mathbf{T}_τ is V -1-Lipschitz and commutes with shifts up to enriched natural isomorphism. *Use:* stability of dashboards and Overlap Gate tolerances.

F.B. AWFS skeleton for C_τ (Spec). Postulate an algebraic weak factorization system $(\mathcal{L}_\tau, \mathcal{R}_\tau)$ on $\text{FiltCh}(k)$ with functorial factorization $F \xrightarrow{\ell_\tau} C_\tau F \xrightarrow{r_\tau} F$, where ℓ_τ is a τ -collapse cofibration (acyclic at persistence level after \mathbf{T}_τ) and r_τ is a τ -local fibration. *Use:* functoriality of C_τ , pullbacks along \mathcal{R}_τ -maps, and stability of f.q.i. under collapse.

F.C. o-minimal Čech and Iwasawa control (Spec). Let $\mathcal{U} = \{U_i\}$ be a definable right-open cover of a window W with finite Leray depth. Then the Čech complex $\check{C}(\mathcal{U})$ computes window-local Ext^1 (under amplitude ≤ 1) and patches across overlaps. An *Iwasawa control catalyst* is a tuple (Γ, ρ, ν) with a directed index Γ , a non-expansive control $\rho : \Gamma \rightarrow \mathbb{R}_{\geq 0}$, and a compatibility map ν such that the windowed diagnostics are ρ -Cauchy and tail-identify with the apex. *Use:* Overlap Gate—uniform control across towers implies stability-band persistence and global gluing.

F.18. Convergence Manager (DP windows) [Spec]

We wrap Restart/Summability (Appendix D) into a Convergence Manager for dynamic-programming (DP) windowing.

Specification C.9 (DP-Convergence Manager). Maintain, per degree i : current safety margin $\text{gap}_k(i)$, per-window budget $\Sigma\delta_k(i)$, and retention factor $\kappa \in (0, 1]$. On transition $k \rightarrow k + 1$,

$$\text{gap}_{k+1}(i) \leftarrow \max\{0, \kappa(\text{gap}_k(i) - \Sigma\delta_k(i))\}.$$

Accept window k if $\text{gap}_k(i) > \Sigma\delta_k(i)$ and DiagZero ; declare convergence on an interval if acceptance persists across the MECE chain and $\sum_k \Sigma\delta_k(i) < \infty$.

Lean 4 tactic skeleton.

```
namespace AK.Gluing
open AK.Core
meta def converge_windows! :
   $\Pi (\kappa : \mathbb{R}_{\geq 0}) (\deg : \mathbb{Z}) (\text{gaps budgets} : \text{List } \mathbb{R}_{\geq 0}),$ 
  tactic (List  $\mathbb{R}_{\geq 0}$ ) := by admit
-- Intended behavior: compute next gaps via  $\kappa \cdot (\text{gap} - \text{budget})^+$  and
-- produce certificates that  $\text{BGATE}^+$  holds on accepted windows.
end AK.Gluing
```

Coq hint database (sketch).

Create HintDb DPconvergence.
Hint Resolve restart_ok summable_budget : DPconvergence.
(* A Ltac 'converge_windows' computes $\kappa \cdot (\text{gap} - \Sigma \delta)^+ +$ and applies the hints. *)

F.19. Tactic stubs: cnf!, ext1_hom!, reverse_bridge!, converge_windows!

Lean 4 (meta-level).

```
namespace AK.Tactics
open AK.Core AK.Tower AK.LocalEquiv

/-- Put a morphism into CNF after  $T_\tau$  and read  $(, u)$ . -/
meta def cnf! ( $\tau : \mathbb{R}_{\geq 0}$ ) : tactic Unit :=
  `[refine (AK.Core.cnf_after_T $\tau$  _ _ ?f  $\tau$ ); all_goals admit]

/-- Solve goals  $\text{Ext}^1(A, k) \cong \text{Hom}(H^{-1}\{A, k\})$  for  $A \in D^{\wedge}[-1, 0]$ . -/
meta def ext1_hom! : tactic Unit :=
  `[apply AK.Core.ext1_edge; try { exact  $\langle \_ \rangle$  }]

/-- Local reverse bridge under  $E_1(W)=0$  (definable window, amplitude  $\leq 1$ ). -/
meta def reverse_bridge! : tactic Unit := `
  [ apply AK.LocalEquiv.local_reverse_bridge; all_goals admit ]

/-- DP-window convergence manager (Appendix D/E). -/
meta def converge_windows! := AK.Gluing.converge_windows!
end AK.Tactics
```

Coq (Ltac skeleton).

```
Ltac cnf  $\tau$  :=
  (* normalize  $T_\tau f$  to barcode CNF and compute  $(, u)$  *) idtac.

Ltac ext1_hom :=
  (* reduce  $\text{Ext}^1(A, k)$  to  $\text{Hom}(H^{-1}\{A, k\})$  when  $A \in D^{\wedge}[-1, 0]$  *) idtac.

Ltac reverse_bridge :=
```

(* apply local reverse under $E_1(W)=0$ to deduce $PH_1=0$ from $Ext^1=0$ *) idtac.

Ltac converge_windows :=

(* apply restart_ok and summable_budget to chain windows *) eauto with DPconvergence.

F.20. Completion note

Remark C.10 (Completion note). This appendix delivers IMRN/AiM-ready formalization stubs: T_τ (exact, idempotent, 1-Lipschitz), CNF after truncation to read (μ, u) , tower calculus with invariances and sufficient conditions (S1–S3), the bridge $PH_1 \Rightarrow Ext^1$ and its local reverse under $E_1(W) = 0$, spectral auxiliaries (non-expansive low-pass with filter-scale σ), and a DP-style Convergence Manager for MECE window pasting. Lean/Coq tactic skeletons (cnf!, ext1_hom!, reverse_bridge!, converge_windows!) provide an auditable path from the paper’s hypotheses to machine-checked goals. All claims are confined to the constructible, amplitude- ≤ 1 regime with filtered colimits used only under the scope rule. No further supplementation is required for operational use in the proof framework.

Appendix G. Reproducibility: Logs and Schemas [Spec] (reinforced)

This appendix specifies the provenance log (run.yaml) and the machine-readable schemas for artifacts produced in this work—barcodes (bars), spectral indicators (spec), Ext-tests (ext), tower comparison maps (phi), and the windowed length-spectrum audit (Lambda_len). All files may be emitted in either JSON or HDF5; JSON keys coincide with HDF5 group/dataset names. Filtered colimits are used only under the scope policy (Appendix A, Remark .67). Type labels follow *Type I–II / Type III / Type IV*. Cross-module conventions: the Ext-test is always against $k[0]$, i.e. $Ext^1(\mathcal{R}(C_\tau F), k) = 0$ (with C_τ understood up to f.q.i. on $Ho(FiltCh(k))$); the energy exponent satisfies $\alpha > 0$ (default $\alpha = 1$). Spectral monotonicity is asserted only for *deletion-type* operations (Dirichlet/principal/Loewner), with directions fixed by Appendix E; inclusion-type operations are used solely with stability bounds.

Mandatory order (After-Collapse). All comparisons follow the mandatory order:

$$\boxed{\text{for each } t \implies \text{apply } \mathbf{P}_t \implies \text{apply } \mathbf{T}_\tau \implies \text{compare in } \text{Pers}_k^{\text{ft}}}.$$

Declaration D.1 (run.yaml schema: mandatory kernel + controlled extensions [Spec]). **Normative status.**

This appendix is normative: the run.yaml manifest is a *proof object* for auditability (Chapter 17) and is synchronized with Chapter 12 (Decl. 11.1). Accordingly, the *kernel blocks* below are **mandatory** whenever a run makes any quantitative claim.

Canonical keys and alias policy (for backward compatibility). The *canonical* (preferred) keys are those listed in the kernel below. For backward compatibility with older drafts and external tools, the following aliases *may* appear, but **MUST** be treated as synonyms and **MUST** be normalized to canonical keys during verification:

definable.o_minimal_structure	\equiv	definable.structure,
layered_delta.delta_Gal/delta_Tr/delta_Fun	\equiv	layered_delta.deltaGal/deltaTr/deltaFun,
iwasawa.control_finite_bounds.kernel_leq/cokernel_leq	\equiv	kernel_le/cokernel_le,
awfs_2cell.*	\equiv	awfs.*.

Use of aliases is *discouraged*: canonical keys should be used in this manuscript and all released artifacts.

Mandatory kernel blocks. The following blocks (and minimal keys) are mandatory for auditability:

```
quantale:
  name: "[0,inf]_plus"    # or "max-plus", "product", ...
  op: "+"                 # value-level monoid op (aggregator for distances/budgets)
  unit: 0
  order: "<="
  mode: "standard"       # "standard" | "probabilistic" | "product" (Ch.12)
```

```
layered_delta:
  deltaGal: ...          # geometric-algebraic ( $\delta^{\{\text{Gal}\}}$ )
  deltaTr: ...           # discretization/rounding/truncation ( $\delta^{\{\text{Tr}\}}$ )
  deltaFun: ...          # functorial/commutation residuals ( $\delta^{\{\text{Fun}\}}$ )
```

```
definable:
  structure: "R_an,exp"   # or "Denef-Pas"
  window_formulae:
    - "u <= t < u'"      # right-open windows; finite or countable MECE cover
```

```
iwasawa:
  tower_level: ...
  control_finite_bounds: { kernel_le: ..., cokernel_le: ... }
```

```
awfs:
  enabled: true
  two_cell_bounds: ...    # scalar or structured value; see below
```

```
overlap_checks:
  local_equiv: true
  cech_ext1_ok: true
  stability_band_ok: true
```

```
spectral_policy:
  order: "ascending"
  norm: "op"             # or "fro"
spectral_bounds:
  lambda_min: ...
  lambda_max: ...
  lip_tol: ...           # optional if unused
```

```
persistence:
  PH1_zero: true
  Ext1_zero: true
  mu: ...
  nu: ...
  phi_iso_tail: ...
```

```
budget:
```

```
sum_delta: ...          # quantale-valued; may be scalar or structured
safety_margin: ...
gap_tau: ...            # mandatory when Restart/Summability is invoked
```

```
ab_test:
  eta: ...
  policy: ...
  fallback: ...
```

Controlled optional extensions (recommended when used). The following blocks are optional but, if present, become *auditable obligations*:

```
pfbc:
  policy: "after_collapse"      # mandatory when PF/BC is used
  residual_ledger: ["disc","meas"] # charging policy (Appendix N, L)
```

```
restart_summability:
  kappa_min: ...
  sum_delta_bound: ...
```

```
tropical:
  bins: { width: ..., range: [a,b] } # diagnostic binning for aux-bars
```

```
policy:
  after_collapse_only: true
  windows: "right-open"
```

```
# NOTE on awfs.two_cell_bounds:
# may be a scalar bound, or a structured map, e.g.
# two_cell_bounds: { mirror_collapse: 0.005, transfer_collapse: 0.005 }
```

Version note (suite v17.0). This version integrates: (i) window declarations and coverage checks, (ii) operation logs with δ -breakdowns, (iii) Overlap Gate checks, (iv) Λ_{len} (length spectrum audit) cross-linked from all artifacts, (v) canonical spectral policy (ascending order; declared norm; bounded safe low-pass flags), (vi) Restart/Summability bookkeeping via `gap_tau` when invoked, and (vii) HDF5 canonicalization with fixed-length UTF-8 strings. These fields ensure replayability and third-party auditability.

G.1. Provenance, determinism, and gating

Each run records (i) source/inputs, (ii) algorithmic choices and thresholds, (iii) numeric tolerances and units, (iv) code/environment fingerprints, (v) RNG details, (vi) strong identifiers (content hashes) for all artifacts, and (vii) *gating* decisions that determine acceptance of results. Randomness is controlled by explicit seeds. Floating-point claims report both an *asserted* tolerance and a *measured* slack. Windows (domain/collapse/spectral) must be declared, and a coverage check attests that all measured quantities fall inside their stated windows. A budget aggregates operation-level error contributions δ and yields a safety margin relative to the governing tolerance; finally, `gate.accept` records the run-level decision (accept/reject) together with reasons.

G.2. run.yaml schema (versions, windows, overlap, budget, gate)

Intent. A single file per execution, sufficient to reproduce the pipeline end-to-end, including all windows, coverage checks, operation logs, overlap checks, and the final acceptance gate.

Canonical layout (YAML).

```
version: 17
schema_version: "2025-03-15"
suite_version: "v17.0"
run_id: "2025-03-15T09:12:07Z-7f5c1b1"
```

```
# Recommended: record coefficients explicitly (must agree with bars/ext artifacts).
coeff_field: "k"           # e.g. "k"; at [Spec] may be "Novikov(q)" etc.
```

```
seed: 1337
rng:
  python: "default_rng"
  numpy: "PCG64"
```

```
platform:
  os: "Ubuntu 22.04"
  cpu: "Intel(R) Xeon(R) Platinum 8370C"
  cuda: "12.2"
  blas: "OpenBLAS 0.3.23"
  hdf5: "1.14.3"
  lapack: "OpenBLAS-LAPACK"
  glibc: "2.35"
  kernel: "5.15.0-105"
  locale: "C.UTF-8"
```

```
env:
  python: "3.11.7"
  packages:
    numpy: "1.26.4"
    scipy: "1.13.1"
    h5py: "3.10.0"
    networkx: "3.2.1"
  threads:
    OMP_NUM_THREADS: 1
    MKL_NUM_THREADS: 1
    OPENBLAS_NUM_THREADS: 1
```

```
container:
  image: "docker.io/example/persistence:2025.03"
  digest: "sha256:deadbeef..."
```

```
git:
  repo: "git@host:ak/persistence.git"
  commit: "a1b2c3d4"
```

```

units:
  filtration: "dimensionless"
  eigenvalues: "dimensionless"

# =====
# ---- Mandatory kernel blocks (Ch.12, Dec. dec:12-schema) ----
# =====
quantale:
  name: "[0,inf]_plus"      # examples: "[0,inf]_plus", "max-plus", "product"
  op: "+"                  # aggregator used for budgets/residuals
  unit: 0.0
  order: "<="
  mode: "standard"        # optional enumerator; used by auditors

layered_delta:
  deltaGal: 0.020
  deltaTr: 0.015
  deltaFun: 0.015

definable:
  structure: "R_an,exp"    # or "Denef-Pas"
  window_formulae: ["u <= t < u'"]
  # optional unless overlap/gluing uses Čech controls:
  cech_depth_bound: 2

iwasawa:
  tower_level: 128
  control_finite_bounds:
    kernel_le: 2
    cokernel_le: 0

awfs:
  enabled: true
  two_cell_bounds:
    mirror_collapse: 0.005
    transfer_collapse: 0.005

# Spectral policy is declared once here (Stage params must agree).
spectral_policy:
  order: "ascending"
  norm: "fro"             # "fro" or "op"
spectral_bounds:
  lambda_min: 1.0e-12
  lambda_max: 1.0e+05
  lip_tol: 0.02           # optional; omit if not used

windows:

```

```

domain:
  filtration_range: [0.0, 2.0]
  degrees: [0, 2]
collapse:
  tau_sweep: [0.25, 0.50, 1.00]
spectral:
  range: [0.0, 2.0]
  order: "ascending"

coverage_check:
  domain_window_coversBars: true
  spectral_window_coversThetas: true
  collapse_tau_sweep_coversReports: true

stability_bands:
  - { i: 1, tau_lo: 0.60, tau_hi: 0.95 } # Appendix D.8

overlap_checks:
  local_equiv: true
  cech_ext1_ok: true
  stability_band_ok: true

# Ch.12: A/B mandatory
ab_test:
  eta: 0.01
  policy: "soft-commuting"
  fallback: "A_then_B"

# =====
# Inputs and pipeline (auditable; stage params must be explicit)
# =====
inputs:
  dataset: "AK-bench-v3"
  graphs:
    - path: "data/G_001.edgelist"
      hash: "sha256:..."
  filters:
    type: "height"
    params: { axis: 2 }

pipeline:
  metric: "interleaving" # exactly one of: interleaving | bottleneck
  stages:
    - name: "barcode"
      params:
        field: "k" # must agree with coeff_field
        reduction: "clearing"
    - name: "collapse" #  $C_\tau$  (up to f.q.i.)

```

```

  params: { tau: 0.50 }
- name: "spec"
  params:
    window: [0.0, 2.0]      # right-open per policy.windows
    # Must agree with spectral_policy:
    norm: "fro"
    order: "ascending"
    # Bounds must agree with top-level spectral_bounds:
    spectral_bounds: { lambda_min: 1.0e-12, lambda_max: 1.0e+05, lip_tol: 0.02 }
    clip: 1.00
  loewner_assumption: "Aprime_preceq_A" # Aprime_preceq_A | Aprime_succeq_A | none
  low_pass:
    kernel: "heat"
    even: true
    mass: 1.0
    clip_tau_eq_pipeline: true
  eig_solver:
    method: "lanczos"
    k: 128
    maxiter: 1000
    tol: 1e-12
    reorthogonalize: true
    rng_seed: 1337
- name: "ext-test"          #  $\text{Ext}^1(R(C_\tau F), k)$ 
  params: { amplitude_check: true }

operations:
- step: 1
  U: [0,1,3]
  type: "inclusion"
  tau: 0.50
  delta:
    distance: { interleaving: 0.050 }
  sources:
    discretization: 0.030
    rounding: 1.0e-12
    heuristic: 0.020
  total: 0.050
  note: "Edge contraction in subgraph U"

persistence:
  PH1_zero: true
  Ext1_zero: true
  mu: 1
  nu: 0
  phi_iso_tail: false      # run-level summary flag (details in phi artifact)

spectral:

```

```

auxBarsRemaining: 0

thresholds:
  alpha: 1.0
  tol:
    distance:
      interleaving: 1e-6
    eig: 1e-8
    witness: 1e-9

# Ch.12: Budget mandatory (quantale-sum target)
budget:
  sum_delta: 0.150
  safety_margin: 0.850
  gap_tau: 0.025
  rationale: "All deltas accounted for; slack remains >0"

# Ch.12: Length spectrum audit mandatory (T15)
LambdaLen:
  degree: 1
  tau: 0.50
  audit: "hash:2f4c...d1"

gate:
  accept: true
  reason: "Coverage ok; safety margin positive; all assertions satisfied"

# =====
# Optional but auditable obligations (present iff used)
# =====
pfbc:
  policy: "after_collapse"      # enforces T-PFBC-AfterCollapse order
  residual_ledger: ["disc", "meas"]

restart_summability:
  kappa_min: 0.8
  sum_delta_bound: 0.05

tropical:
  bins: { width: 0.02, range: [0.0, 2.0] }

# If navigation uses gradients (Ch.13-15), this block is mandatory:
grad_policy:
  method: "finite_difference"   # finite_difference | SPSA | surrogate
  norm: "fro"                  # norm for step decisions
  stencil: { kind: "two_sided", eps: 1.0e-3, coords: "all" }
  seed: 1337
  variance_reporting: "estimate" # estimate | upper_bound

```

```

delta_charge_target: "operations[*].delta.sources.heuristic"

policy:
  after_collapse_only: true
  windows: "right-open"

serialization:
  float_dtype: "ieee754-f64-le"
  json_sort_keys: true
  hdf5_canonical:
    compression: { algo: "gzip", level: 4 }
    shuffle: false
    fletcher32: false
    track_times: false
    fillvalue: 0.0
    string_encoding: "utf8-fixed"
  chunk_shapes:
    bars: { i: 4096, birth: 4096, death: 4096, death_is_inf: 4096, mult: 4096 }
    spec_eigs: { eig: 4096 }
    spec_Ntheta: { theta: 512, left: 512, right: 512 }
    phi_idx: { i: 256, tau: 256, iso: 256, mu: 256, nu: 256 }

status:
  success: true
  errors: []

outputs:
  bars: "out/bars_7f5c1b1.json"
  spec: "out/spec_7f5c1b1.json"
  ext: "out/ext_7f5c1b1.json"
  phi: "out/phi_7f5c1b1.h5"
  Lambda_len: "out/Lambda_len_7f5c1b1.json"

```

G.3. bars (barcodes) schema

Semantics. A constructible barcode is a multiset of half-open intervals $I = [b, d)$ with degree i . Deaths may be $+\infty$.

JSON layout (infinity convention, units, cross-links, optional clip report).

```

{
  "meta": {
    "schema_version": "2025-03-15",
    "suite_version": "v17.0",
    "field": "k",
    "filtration_units": "dimensionless",
    "endpoint_convention": "[b,d) (see Chapter 2)",
    "infinity": { "json": "inf" },
    "clip_tau": 0.50,

```

```

    "float_dtype": "ieee754-f64-le",
    "string_encoding": "utf8-fixed",
    "links": {
      "run_id": "2025-03-15T09:12:07Z-7f5c1b1",
      "run_yaml_hash": "sha256:...run",
      "Lambda_len": "sha256:...Lambda"
    }
  },
  "bars": [
    { "i": 0, "birth": 0.0, "death": 0.3, "mult": 1 },
    { "i": 1, "birth": 0.2, "death": "inf", "mult": 1 }
  ],
  "hash": "sha256:...bars"
}

```

HDF5 layout (split representation for $+\infty$; fixed UTF-8).

- **Datasets:** /bars/i (int32), /bars/birth (float64), /bars/death (float64), /bars/death_is_inf (bool), /bars/mult (int32).
- **Attributes:** /bars.attrs[field="k"], filtration_units, schema_version, suite_version, float_dtype, death_encoding="split_scalar_bool", string_encoding="utf8-fixed", optional clip_tau, and links/Lambda_len.

G.4. spec (spectral indicators) schema

Semantics. Spectral features include: clipped sums, counts above/below thresholds with left/right limits, and deletion-type monotonicity diagnostics (Appendix E). Matrices/operators are identified by content hashes. All spectral reporting is post-collapse unless explicitly marked otherwise by policy.

JSON layout (ascending storage; $N_{\theta \pm 0}$; solver and low-pass params; coverage; cross-links).

```

{
  "meta": {
    "schema_version": "2025-03-15",
    "suite_version": "v17.0",
    "eigen_units": "dimensionless",
    "order": "ascending",
    "sorted": true,
    "Ntheta_convention": { "left": "N_{\theta-0}", "right": "N_{\theta+0}" },
    "window": { "range": [0.0, 2.0], "semantics": "right-open" },
    "norm": "fro",
    "clip_tau": 1.0,
    "tol_eig": 1e-8,
    "loewner_assumption": "Aprime_preceq_A",
    "low_pass": { "kernel": "heat", "even": true, "mass": 1.0, "safe": true },
    "aux_bars_remaining": 0,
    "coverage_check": { "thetas_in_window": true },
    "tropical_bins": { "width": 0.02, "range": [0.0, 2.0] },
    "string_encoding": "utf8-fixed",
    "eig_solver": {

```

```

    "method": "lanczos", "k": 128, "maxiter": 1000,
    "tol": 1e-12, "reorthogonalize": true, "rng_seed": 1337
  },
  "links": {
    "run_id": "2025-03-15T09:12:07Z-7f5c1b1",
    "run_yaml_hash": "sha256:...run",
    "Lambda_len": "sha256:...Lambda"
  }
},
"operators": [
  {
    "id": "sha256:...A",
    "kind": "laplacian_dirichlet",
    "n": 500,
    "spectrum": { "eigs": [0.10, 0.12, 0.45, ...] },
    "clip": { "tau": 1.0, "sum": 37.219, "deficit": 12.004 }
  },
  {
    "id": "sha256:...B",
    "kind": "principal_submatrix",
    "parent": "sha256:...A",
    "N_theta": [
      { "theta": 0.20, "left": 17, "right": 16 },
      { "theta": 0.50, "left": 10, "right": 10 }
    ],
    "monotonicity": { "type": "deletion", "passed": true }
  }
],
"hash": "sha256:...spec"
}

```

HDF5 layout.

- /spec/ops/{id}/eig (float64, ascending), /spec/ops/{id}/clip/sum (float64), /spec/ops/{id}/clip/deficit (float64), /spec/ops/{id}/Ntheta/theta, /spec/ops/{id}/Ntheta/left, /spec/ops/{id}/Ntheta/right (parallel datasets).
- Attributes: kind, parent, norm $\in \{ "fro", "op" \}$, order="ascending", sorted (bool), eigen_units, tol_eig, schema_version, suite_version, loewner_assumption, low_pass/*, auxBarsRemaining, coverage_thetas_in_window, optional tropical_bins/width, range, string_encoding="utf8-fixed", and links/Lambda_len.

G.5. ext (Ext-test) schema

Semantics. Outcome of the bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ for $C_\tau F$, with amplitude checks for \mathcal{R} and recorded assumptions.

JSON layout (assumptions, cross-links).

```

{
  "meta": {

```

```

"schema_version": "2025-03-15",
"suite_version": "v17.0",
"field": "k",
"alpha": 1.0,
"assumptions": {
  "field_is_k": true,
  "constructible_verified": true,
  "t_exact_and_amp_le_1": true
},
"string_encoding": "utf8-fixed",
"links": {
  "run_id": "2025-03-15T09:12:07Z-7f5c1b1",
  "run_yaml_hash": "sha256:...run",
  "Lambda_len": "sha256:...Lambda"
}
},
"tau": 0.50,
"amplitude": { "ok": true, "range": [-1, 0] },
"Hminus1": { "dim": 0, "witness_norm": 0.0 },
"Ext1": { "dim": 0, "passed": true, "tol": 1e-9, "slack": 0.0 },
"links": { "bars": "sha256:...bars", "phi": "sha256:...phi" },
"hash": "sha256:...ext"
}

```

HDF5 layout.

- Scalars: /ext/tau (float64), /ext/Hminus1/dim (int32), /ext/Ext1/dim (int32), /ext/Ext1/passed (bool), /ext/Ext1/tol (float64), /ext/Ext1/slack (float64).
- Attributes: field="k", alpha (float64), schema_version, suite_version, assumptions/*, string_encoding="utf8-fixed", and links/Lambda_len.

G.6. phi (tower comparison) schema

Semantics. Encodes $\phi_{i,\tau}$ for towers, together with (μ, u) as generic-fiber dimensions after truncation, structural flags for the sufficiency hypotheses (Appendix D, §D.4), and explicit stability bands (Appendix D.8).

JSON layout (with τ -sweep, stability bands, witnesses, iso-tail, cross-link).

```

{
  "meta": {
    "schema_version": "2025-03-15",
    "suite_version": "v17.0",
    "definition": "phi_{i,\tau}: colim T_\tau P_i(F_n) \to T_\tau P_i(F_\infty)",
    "scope": "colim in [\mathbb{R}, Vect_k], return-to-constructible policy",
    "tau_sweep": [0.25, 0.50, 1.00],
    "edge_kinds": ["inclusion", "projection", "quasi_iso",
      "filtration_preserving_map", "schur_complement", "other"],
    "stability_bands": [ { "i": 1, "tau_lo": 0.60, "tau_hi": 0.95 } ],
    "string_encoding": "utf8-fixed",

```

```

"links": {
  "run_id": "2025-03-15T09:12:07Z-7f5c1b1",
  "run_yaml_hash": "sha256:...run",
  "Lambda_len": "sha256:...Lambda"
},
"indices": [
  {
    "i": 1, "tau": 0.50,
    "iso": false,
    "mu": 1, "nu": 0,
    "flags": { "S1_commutes": false, "S2_noAccum": true, "S3_Cauchy": false },
    "witness": { "ker_generic_dim": 1, "coker_generic_dim": 0 },
    "iso_tail": { "passed": false }
  }
],
"tower": {
  "nodes": [
    { "n": 0, "id": "sha256:...F0" },
    { "n": 1, "id": "sha256:...F1" }
  ],
  "edges": [
    { "src": 0, "dst": 1, "kind": "inclusion" }
  ],
  "limit": { "id": "sha256:...Finf" }
},
"hash": "sha256:...phi"
}

```

HDF5 layout.

- /phi/idx/i (int32), /phi/idx/tau (float64), /phi/idx/iso (bool), /phi/idx/mu (int32), /phi/idx/nu (int32).
- /phi/idx/flags/S1_commutes, /phi/idx/flags/S2_noAccum, /phi/idx/flags/S3_Cauchy (bool).
- Optional witnesses: /phi/idx/witness/ker_generic_dim, /phi/idx/witness/coker_generic_dim.
- Optional tail: /phi/idx/iso_tail/passed (bool).
- /phi/meta/stability_bands: records $(i, \tau_{lo}, \tau_{hi})$.
- Optional tower edges: /phi/tower/edges/src, /phi/tower/edges/dst (int32), /phi/tower/edges/kind (fixed-length UTF-8 string).
- Attributes: schema_version, suite_version, string_encoding="utf8-fixed", tau_sweep (float64 array), and links/Lambda_len.

G.7. Lambda_len (windowed length spectrum) schema

Semantics. The length spectrum operator $\Lambda_{\text{len}}(M; [0, \tau])$ is diagonal on the bar-basis with eigenvalues equal to clipped bar-lengths on $[0, \tau]$. Its unordered eigenvalue multiset equals the clipped bar-length multiset (Appendix H). The Lambda_len audit records either the eigenvalue list (small instances) or a content hash.
JSON layout (links to all major artifacts).

```
{
  "meta": {
    "schema_version": "2025-03-15",
    "suite_version": "v17.0",
    "definition": "Lambda_len(T_tau P_i(C_tau F); [0,tau])",
    "degree": 1,
    "tau": 0.50,
    "string_encoding": "utf8-fixed",
    "links": {
      "bars": "sha256:...bars",
      "phi": "sha256:...phi",
      "spec": "sha256:...spec",
      "ext": "sha256:...ext",
      "run_yaml_hash": "sha256:...run"
    }
  },
  "eigs": [0.24, 0.51, 0.78],
  "hash": "sha256:2f4c...d1"
}
```

HDF5 layout.

- /Lambda_len/meta attributes: schema_version, suite_version, degree, tau, string_encoding="utf8-fixed", and links as above.
- /Lambda_len/eigs (optional; float64 array).
- /Lambda_len/hash (fixed-length UTF-8 string).

G.8. Content hashing and canonical serialization

Each artifact carries a content hash sha256:... over its canonical serialization (JSON with sorted keys; HDF5 with fixed dataset/attribute creation order, chunk shapes, compression and filters). Cross-file links (bars \leftrightarrow phi \leftrightarrow ext \leftrightarrow spec \leftrightarrow Lambda_len) use these hashes exclusively. *JSON numeric policy:* finite numbers only; positive infinity is encoded as the string "inf" where applicable (see bars.meta.infinity). *HDF5:* encodes $+\infty$ via the split representation /bars/death (float64) + /bars/death_is_inf (bool). *Strings:* all JSON strings and HDF5 string datasets/attributes are fixed-length UTF-8 (string_encoding="utf8-fixed") to ensure bitwise reproducibility. *HDF5 canonicalization:* set track_times=false, shuffle=false, fletcher32=false, fillvalue=0.0, compression to GZIP level 4, and use the chunk_shapes recorded in run.yaml; create datasets and attributes in the order shown in this appendix.

G.9. Numeric tolerances, δ -budgets, and audit trail

Every quantitative claim includes:

- **tolerance** (tol) declared in run.yaml;
- **slack** (slack) measured margin to the decision boundary;
- **norm** used for spectral bounds (fro or op), consistent with Appendix E;
- **metric** for persistence distances (interleaving or bottleneck);
- **budget aggregation:** operations[*].delta entries quantale-sum into budget.sum_delta, with budget.safety_margin and budget.gap_tau;
- **layered δ -ledger:** the stratified entries layered_delta.* must sum to the aggregate via quantale.op;
- **solver** details for spectral computations (Lanczos parameters, RNG seed);
- **windows/coverage:** windows.* declare scopes; coverage_check.* record pass/fail;
- **gate decision:** gate.accept with gate.reason.

G.10. Tests T14/T15 and reproducibility checklist

T14 (Overlap Gate gluing). On a windowed cover, verify: (i) post-collapse equality (up to budget) on overlaps (overlap_checks.local_equiv=true), (ii) Čech–Ext¹ acyclicity on overlaps (cech_ext1_ok=true), (iii) stability band detection ($\mu = u = 0$, stability_band_ok=true), (iv) A/B soft-commuting with logged residuals (added to the δ -ledger), (v) global gate acceptance with additive budgets.

T15 (Length spectrum audit). Compute $\Lambda_{\text{len}}(\mathbf{T}_\tau \mathbf{P}_i(C_\tau F); [0, \tau])$ and verify that its eigenvalue multiset equals the clipped bar-length multiset of $\mathbf{T}_\tau \mathbf{P}_i(C_\tau F)$ (Appendix H). Log either the eigenvalue list or a content hash under Lambda_len, and ensure cross-links (bars/spec/ext/phi) resolve to the same hash.

Minimal reproducibility checklist.

1. Preserve run.yaml and all emitted bars/spec/ext/phi/Lambda_len files (JSON or HDF5).
2. Confirm $\alpha > 0$ (default $\alpha = 1$) and field k are consistent across files.
3. Verify content hashes and all cross-links resolve; each artifact carries meta.links.run_id and run_yaml_hash; all artifacts provide a link to Lambda_len.
4. Check that pipeline.metric matches thresholds.tol.distance (exactly one of interleaving/bottleneck); norms/tolerances are consistent across spec.
5. Verify declared windows and that coverage_check.* is true; for definable, check o_minimal_structure, window_formulae, and cech_depth_bound.
6. Verify eigenvalue order metadata: spec.meta.order="ascending", spec.meta.sorted=true, and HDF5 eigen arrays are non-decreasing; if tropical_bins are present, confirm width/range match tropical.bins.
7. For Dirichlet Laplacians, confirm $\lambda_{\min} > 0$ (or log a certified lower bound consistent with Appendix E).
8. Ensure each recorded θ used for N_θ lies within spec.meta.window.range; recompute spectral indicators using recorded norm/tolerance/solver settings; check slack ≥ 0 .

9. Re-evaluate $\phi_{i,\tau}$ under the scope policy; confirm (μ, u) match generic-fiber counts (CNF after \mathbf{T}_τ); check `iso_tail.passed`; if `iwasawa` is present, verify `kernel_leq`, `cokernel_leq`. Confirm listed `stability_bands`.
10. For Ext-tests, verify amplitude $[-1, 0]$, the assumption flags in `ext.meta.assumptions`, and $\text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$; check `persistence.Ext1_zero`.
11. Validate HDF5 canonicalization: `chunk shapes`, `compression`, `filters`, `string_encoding="utf8-fixed"`, and `track_times=false`; `death` is represented via the `split float/bool` fields.
12. Inspect operations, budget, and `layered_delta`: the stratified sums match `budget.sum_delta` via `quantale.op`; compute safety margin and `gap_tau`; verify `gate.accept`.
13. If `awfs_2cell.awfs_enabled`, verify recorded two-cell bounds and that `policy.after_collapse_only` holds.

G.11. Extended policy notes (concise)

- **Quantale.** The quantale section fixes the value-level monoid and order used to aggregate distances and budgets; all layered δ entries combine via `quantale.op`.
- **Definable windows.** Right-open windows are specified by first-order formulae in the declared *o-minimal* structure; the Čech depth bound controls gluing on overlaps.
- **Layered δ .** The stratified ledger $(\delta^{\text{Gal}}, \delta^{\text{Tr}}, \delta^{\text{Fun}})$ provides a MECE breakdown that must quantale-sum to the aggregate.
- **Iwasawa control.** Finite bounds for kernel/cokernel dimensions at the recorded tower level certify control of (μ, u) along the tower.
- **AWFS 2-cell.** Optional two-cell bounds certify functoriality/transport of C_τ across mirrors/transfers/pull-backs.
- **Safe low-pass.** The `low_pass` block records kernel parity (even), unit mass, and alignment of clipping with the pipeline; these ensure non-expansiveness (Appendix E).
- **Tropical bins.** Optional diagnostic binning for aux-bars; if present, the same bin policy is mirrored in `spec.meta.tropical_bins`.

Outcome. The versioned schemas above—now with (i) overlap checks for gluing, (ii) a windowed length-spectrum audit, (iii) canonical spectral policy (`order="ascending"`, `norm="op"|"fro"`, spectral bounds with safe low-pass), (iv) δ -ledger extensions with `layered_delta` and `gap_tau`, (v) quantale/definable/Iwasawa/AWFS/tropical hooks, (vi) explicit `stability_bands`, and (vii) HDF5 canonicalization with fixed-length UTF-8 strings—are sufficient to regenerate all figures and claims in the main text from first principles, within the constructible regime and under the filtered-colimit policy, while making `accept/reject` criteria explicit and auditable. No further supplementation is required for operational deployment or third-party review.

Appendix H. Betti Integral and Finite τ -Events (Reinforced)

Standing conventions. We work over a field k . All persistence modules are constructible (locally finite on bounded windows); filtered colimits, when used, are taken only under the scope policy of Appendix A, Remark .67. Endpoint conventions follow Appendix A, Remark .63; we use half-open bars $[b, d)$ (any consistent endpoint convention yields the same integrals, cf. Remark E.3). Global conventions: Ext-tests are always against $k[0]$ (we write $\text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$, with C_τ understood up to f.q.i. on $\text{Ho}(\text{FiltCh}(k))$); the *energy exponent* satisfies $\alpha > 0$ (default $\alpha = 1$). Window and overlap claims, coverage checks, and acceptance gates are recorded in the manifest run.yaml (Appendix D). Type labels follow *Type I–II / Type III / Type IV*.

H.1. Betti curves and the Betti integral

Let F be a filtered chain complex (or any filtered object realizing a persistence module) and let $\mathbf{P}_i(F)$ denote its degree- i persistence module. Write the corresponding barcode decomposition (constructible, locally finite on bounded windows) as a multiset

$$\mathbf{P}_i(F) \cong \bigoplus_{I \in \mathcal{B}_i(F)} I^{\oplus m(I)}, \quad I = [b, d), \quad d \in \mathbb{R} \cup \{\infty\}, \quad m(I) \in \mathbb{Z}_{\geq 1}.$$

Define the Betti curve $\beta_i(t) := \dim_k H_i(F^t)$ and the *Betti integral* up to $\tau \geq 0$ by

$$\text{PE}_i^{\leq \tau}(F) := \int_0^\tau \beta_i(t) dt.$$

Under constructibility, β_i is right-continuous and piecewise constant, and on any bounded window only finitely many bars meet.

Theorem E.1 (Betti integral = clipped barcode mass). *For every $\tau \geq 0$,*

$$\text{PE}_i^{\leq \tau}(F) = \sum_{I \in \mathcal{B}_i(F)} m(I) \cdot \lambda(I \cap [0, \tau]),$$

where λ is Lebesgue measure and

$$\lambda([b, d) \cap [0, \tau]) = \max\{0, \min\{d, \tau\} - \max\{b, 0\}\} \quad (\min\{\infty, \tau\} = \tau).$$

In particular, an infinite bar alive at 0 contributes its clipped length τ .

Proof. By local finiteness, for each bounded window $[0, \tau]$ only finitely many bars intersect the window, so

$$f(t) := \sum_{I \in \mathcal{B}_i(F)} m(I) \mathbf{1}_I(t)$$

is a nonnegative measurable function on $[0, \tau]$ given by a finite sum. With the half-open convention $[b, d)$, for all t away from event times (births b and deaths d) we have $\beta_i(t) = f(t)$; and since β_i is right-continuous, $\beta_i = f$ almost everywhere on $[0, \tau]$. Thus, by Tonelli/Fubini on bounded windows,

$$\int_0^\tau \beta_i(t) dt = \int_0^\tau f(t) dt = \sum_I m(I) \int_0^\tau \mathbf{1}_I(t) dt = \sum_I m(I) \lambda(I \cap [0, \tau]).$$

□

Corollary E.2 (Monotonicity, a.e. derivative, piecewise linearity). *The map $\tau \mapsto \text{PE}_i^{\leq \tau}(F)$ is nondecreasing, continuous, and piecewise linear on every bounded interval. Its derivative satisfies*

$$\frac{d}{d\tau} \text{PE}_i^{\leq \tau}(F) = \beta_i(\tau) \quad \text{for a.e. } \tau,$$

and at event points (births/deaths) the right derivative equals $\beta_i(\tau)$ while the left derivative equals $\beta_i(\tau-)$. All breakpoints on $[0, \tau_0]$ lie in $\{0, \tau_0\} \cup \{b \in [0, \tau_0]\} \cup \{d \in [0, \tau_0]\}$.

Remark E.3 (Endpoint and baseline conventions). Changing open/closed endpoint conventions modifies β_i only on a set of measure zero; the integral and breakpoint set remain unchanged. The baseline 0 is a reference; negative births are allowed and handled by intersecting I with $[0, \tau]$.

Remark E.4 (Energy exponent and α -Betti integral). For $\alpha > 0$, define the α -Betti integral up to $\tau \geq 0$ by

$$\text{PE}_{i,\alpha}^{\leq \tau}(F) := \int_0^\tau (\beta_i(t))^\alpha dt.$$

On each component of $[0, \tau]$ between consecutive event times, β_i is constant; hence $\text{PE}_{i,\alpha}^{\leq \tau}$ is continuous, nondecreasing, and piecewise linear in τ , with slope $(\beta_i(\tau))^\alpha$ on right-open pieces. The case $\alpha = 1$ recovers Theorem E.1. For $\alpha \neq 1$, $\text{PE}_{i,\alpha}^{\leq \tau}$ is not additive over bars (because $x \mapsto x^\alpha$ is nonlinear), but all finite-check and implementation statements below remain valid verbatim (replace β_i by β_i^α when computing segment slopes).

Corollary E.5 (Trace/length-spectrum identity for $\alpha = 1$). *Let $\Lambda_{\text{len}}(\mathbf{P}_i(F); [0, \tau])$ denote the length-spectrum operator of Appendix G (§G.7), diagonal in the bar-basis with diagonal entries $\ell_\tau(I) := \lambda(I \cap [0, \tau])$ repeated with multiplicity $m(I)$. Then*

$$\text{tr}(\Lambda_{\text{len}}(\mathbf{P}_i(F); [0, \tau])) = \sum_I m(I) \ell_\tau(I) = \text{PE}_i^{\leq \tau}(F).$$

Consequently, the Λ_{len} audit (Appendix G) is a trace-level certificate of Theorem E.1 when $\alpha = 1$.

H.2. Finite τ -events and finite checking sets

Fix $\tau_0 > 0$ and define the finite τ -event set

$$\text{Ev}_i(F; \tau_0) := \{0, \tau_0\} \cup (\{b \mid [b, d] \in \mathcal{B}_i(F)\} \cap [0, \tau_0]) \cup (\{d \mid [b, d] \in \mathcal{B}_i(F)\} \cap [0, \tau_0]).$$

By constructibility, $\text{Ev}_i(F; \tau_0)$ is finite.

Proposition E.6 (Finite checking set). *Let $g : [0, \tau_0] \rightarrow \mathbb{R}$ be continuous and affine on each connected component of $[0, \tau_0] \setminus \text{Ev}_i(F; \tau_0)$ (e.g. a piecewise linear benchmark with breakpoints in Ev_i). Then, for either inequality direction,*

$$\text{PE}_i^{\leq \tau}(F) \geq g(\tau) \quad (\text{resp. } \leq) \quad \text{for all } \tau \in [0, \tau_0]$$

holds if and only if it holds for all $\tau \in \text{Ev}_i(F; \tau_0)$.

Proof. Between consecutive event times, both $\text{PE}_i^{\leq \tau}(F)$ and $g(\tau)$ are affine in τ (Corollary E.2). Hence their difference $h(\tau) := \text{PE}_i^{\leq \tau}(F) - g(\tau)$ is affine on each closed component $J = [u, v] \subset [0, \tau_0] \setminus \text{Ev}_i(F; \tau_0)$. An affine function on a compact interval attains its extremum at an endpoint, so $h \geq 0$ (resp. $h \leq 0$) on J iff $h(u) \geq 0$ and $h(v) \geq 0$ (resp. ≤ 0). Taking the union over all such J plus the singleton event points yields the claim. \square

Remark E.7 (Definable piecewise constancy (o-minimal); finite checks). Under the definable windowing policy of Appendix G (Declaration D.1, definable.o_minimal_structure/window_formulae), each β_i on a bounded window is definable and integer-valued, hence piecewise constant with finitely many jumps. Equivalently, $\text{Ev}_i(F; \tau_0)$ is finite and computable from definable data. Consequently, all constraints on $\text{PE}_{i,\alpha}^{\leq \tau}$ that are affine on components between events reduce to finitely many evaluations at $\tau \in \text{Ev}_i(F; \tau_0)$; record the event list (or its hash), window declaration, and counts in the manifest run.yaml (Appendix G).

Remark E.8 (Algorithmic evaluation). Algorithm for evaluating $\text{PE}_{i,\alpha}^{\leq \tau}$ on $[0, \tau_0]$:

1. Collect all births and deaths intersecting $[0, \tau_0]$; sort to form $\text{Ev}_i(F; \tau_0) = \{0 = t_0 < t_1 < \dots < t_M = \tau_0\}$.
2. For each segment $[t_j, t_{j+1})$, compute $c_j := \beta_i(t)$ for any $t \in [t_j, t_{j+1})$ and set $s_j := c_j^\alpha$.
3. For $\tau \in [t_j, t_{j+1})$,

$$\text{PE}_{i,\alpha}^{\leq \tau} = \sum_{\ell < j} s_\ell (t_{\ell+1} - t_\ell) + s_j (\tau - t_j).$$

Complexity: $O(M \log M)$ to form/sort Ev_i and $O(M)$ for accumulation. Record the window definition, endpoint policy (right-open), tie-break rules, and event counts in run.yaml (Appendix G).

H.3. Consequences for shifts, truncations, and window variation

(i) **Baseline/interval form and shifts.** For $a < b$, define the interval Betti integral

$$\text{PE}_{i,\alpha}^{[a,b]}(F) := \int_a^b (\beta_i(t))^\alpha dt.$$

Then $\text{PE}_{i,\alpha}^{\leq \tau}(F) = \text{PE}_{i,\alpha}^{[0,\tau]}(F)$. For $\varepsilon \in \mathbb{R}$, let $(S^\varepsilon F)^t := F^{t+\varepsilon}$. Then for any $a < b$,

$$\text{PE}_{i,\alpha}^{[a,b]}(S^\varepsilon F) = \text{PE}_{i,\alpha}^{[a+\varepsilon, b+\varepsilon]}(F).$$

In particular, when $\varepsilon \geq 0$ and $\sigma \geq 0$,

$$\text{PE}_{i,\alpha}^{\leq \sigma}(S^\varepsilon F) = \text{PE}_{i,\alpha}^{\leq \sigma+\varepsilon}(F) - \text{PE}_{i,\alpha}^{\leq \varepsilon}(F).$$

(ii) **Truncation monotonicity (deletion-type).** Let $\mathbf{T}_{\tau'}$ denote bar-deletion at scale $\tau' > 0$ (Appendix A). Then, for every $\sigma > 0$ and $\alpha > 0$,

$$\text{PE}_{i,\alpha}^{\leq \sigma}(\mathbf{T}_{\tau'}(\mathbf{P}_i(F))) \leq \text{PE}_{i,\alpha}^{\leq \sigma}(\mathbf{P}_i(F)),$$

since β_i decreases pointwise under deletion and $x \mapsto x^\alpha$ is nondecreasing on $\mathbb{R}_{\geq 0}$ for $\alpha > 0$. Moreover, $\mathbf{T}_{\tau'}$ is 1-Lipschitz in interleaving distance (Appendix A).

(iii) **Lipschitz in the window parameter.** For $0 \leq s \leq \tau$ and $\alpha > 0$,

$$|\text{PE}_{i,\alpha}^{\leq \tau}(F) - \text{PE}_{i,\alpha}^{\leq s}(F)| = \int_s^\tau (\beta_i(t))^\alpha dt \leq (\tau - s) \cdot \sup_{t \in [s, \tau]} (\beta_i(t))^\alpha.$$

H.4. Stability under matchings and perturbations on bounded windows

For a window $[a, b]$, write $N_i([a, b])$ for the total number of bars (counted with multiplicity) in $\mathcal{B}_i(F) \cup \mathcal{B}_i(G)$ that intersect $[a, b]$. (Any equivalent locally finite counting convention is acceptable, provided it is fixed and recorded.)

Proposition E.9 (Windowed perturbation bounds). *Fix $\tau_0 > 0$. Suppose two barcodes $\mathcal{B}_i(F)$ and $\mathcal{B}_i(G)$ admit a bottleneck δ -matching on $[-\delta, \tau_0 + \delta]$: each matched pair $[b, d] \leftrightarrow [b', d']$ satisfies $|b - b'| \leq \delta$, $|d - d'| \leq \delta$ (with $d = \infty$ allowed), and every unmatched bar has length $\leq 2\delta$. Then:*

1. ($\alpha = 1$) For all $\tau \in [0, \tau_0]$,

$$|\text{PE}_i^{\leq \tau}(F) - \text{PE}_i^{\leq \tau}(G)| \leq 2\delta \cdot N_i([-\delta, \tau_0 + \delta]).$$

2. ($\alpha \geq 1$) Let $B := \sup_{t \in [-\delta, \tau_0 + \delta]} \max\{\beta_i^F(t), \beta_i^G(t)\}$ (finite by constructibility). Then for all $\tau \in [0, \tau_0]$,

$$|\text{PE}_{i,\alpha}^{\leq \tau}(F) - \text{PE}_{i,\alpha}^{\leq \tau}(G)| \leq 2\alpha B^{\alpha-1} \delta \cdot N_i([-\delta, \tau_0 + \delta]).$$

3. ($0 < \alpha \leq 1$) For all $\tau \in [0, \tau_0]$,

$$|\text{PE}_{i,\alpha}^{\leq \tau}(F) - \text{PE}_{i,\alpha}^{\leq \tau}(G)| \leq 2\delta^\alpha \cdot N_i([-\delta, \tau_0 + \delta]).$$

Proof sketch. For $\alpha = 1$, Theorem E.1 expresses $\text{PE}_i^{\leq \tau}$ as a sum of clipped lengths on $[0, \tau]$. Endpoint perturbations by δ change each matched clipped length by at most 2δ ; each unmatched bar contributes at most 2δ . Summing over all bars intersecting $[-\delta, \tau_0 + \delta]$ yields (1). For $\alpha \geq 1$, use the pointwise bound $|x^\alpha - y^\alpha| \leq \alpha \max\{x, y\}^{\alpha-1} |x - y|$ and the fact that β_i changes only at event times, whose total count on the enlarged window is controlled by $N_i([-\delta, \tau_0 + \delta])$. For $0 < \alpha \leq 1$, use $|x^\alpha - y^\alpha| \leq |x - y|^\alpha$ on $\mathbb{R}_{\geq 0}$ and the same windowed counting argument. \square

Remark E.10 (Metric policy). The manifest run.yaml (Appendix G) records whether the pipeline uses interleaving or bottleneck for distance reporting. Proposition E.9 is stated for a bottleneck-style endpoint control; it applies whenever a bottleneck bound is derived or logged (e.g. by stability reductions or explicit matchings). For interleaving-only runs, use a conversion lemma (if available in the implementation) or log the induced bottleneck bound explicitly.

H.5. Implementation notes and numerics

For large barcodes, the following practices improve reproducibility and numerical stability:

1. **Event extraction:** derive $\text{Ev}_i(F; \tau_0)$ directly from the barcode; for streamed persistence, emit birth-s/deaths as they occur and maintain a running count c_j .
2. **Accumulation:** use compensated summation (e.g. Kahan) when aggregating $s_j (t_{j+1} - t_j)$.
3. **Types:** store event times as float64; store counts c_j as 64-bit integers; compute slopes as float64.
4. **Idempotence:** with $[b, d]$, repeated evaluation on the same event sequence is bitwise deterministic (fixed sort and tie-break rules on equal times).
5. **Window policy:** record in run.yaml the baseline, window $[0, \tau_0]$, endpoint convention (right-open), and whether negative births are present (Appendix G).

H.6. Testing and validation

Minimal tests:

1. **Synthetic bars:** verify Theorem E.1 and cross-check numerical integration vs. clipped-length summation ($\alpha = 1$).
2. **Endpoint consistency:** switch between $[b, d)$ and any other consistent convention and verify identical $\text{PE}_{i,\alpha}^{\leq \tau}$ and breakpoint sets.
3. **Shift equivariance:** random ε ; check $\text{PE}_{i,\alpha}^{[a,b]}(S^\varepsilon F) = \text{PE}_{i,\alpha}^{[a+\varepsilon, b+\varepsilon]}(F)$.
4. **Truncation monotonicity:** apply $\mathbf{T}_{\tau'}$; verify pointwise decrease in σ .
5. **Stability:** simulate δ -endpoint perturbations; confirm Proposition E.9 (casework by α).
6. **Length-spectrum audit:** for $\alpha = 1$, verify Corollary E.5 and the multiset identity logged in `Lambda_len` (Appendix G).

H.7. Variants and generalizations

- **Weighted windows.** For nonnegative $w \in L^1([0, \tau_0])$, define

$$\text{PE}_{i,\alpha}^w(F) := \int_0^{\tau_0} w(t) (\beta_i(t))^\alpha dt.$$

If w is piecewise constant with breakpoints contained in $\text{Ev}_i(F; \tau_0)$, then $\text{PE}_{i,\alpha}^w$ reduces to a finite sum over event intervals. For $\alpha = 1$, one additionally has the barcode-level identity

$$\text{PE}_i^w(F) = \sum_{I \in \mathcal{B}_i(F)} m(I) \int_{I \cap [0, \tau_0]} w(t) dt,$$

by linearity of $\beta_i(t) = \sum_I m(I) \mathbf{1}_I(t)$.

- **Alternate baselines.** Replacing $[0, \tau]$ by $[a, b]$ yields (for $\alpha = 1$)

$$\text{PE}_i^{[a,b]}(F) = \sum_I m(I) \lambda(I \cap [a, b]),$$

and for general $\alpha > 0$ one uses $\text{PE}_{i,\alpha}^{[a,b]}(F) = \int_a^b (\beta_i(t))^\alpha dt$.

- **Discrete filtrations.** Replace integrals by Riemann sums on a grid; all statements adapt with counting measure and the event set replaced by grid points.

H.8. E_1 -level determinacy and bridge notes (window-local) [Spec]

Determinacy of $E_1 = 0$ on finite τ -events. In the Overlap Gate, any constraint of the form

$$\text{PE}_{i,\alpha}^{\leq \tau}(F) \leq g(\tau) \quad \text{or} \quad \geq g(\tau)$$

with g continuous and affine on components between event times holds on $[0, \tau_0]$ iff it holds at the finite set $\tau \in \text{Ev}_i(F; \tau_0)$ (Proposition E.6). Under the definable windowing policy (Appendix G, Declaration D.1), $\text{Ev}_i(F; \tau_0)$ is computable and auditable (Remark E.7). Record the event list (or hash), window declaration, and pass/fail results in `run.yaml` (Appendix G).

Bridge note (field/cohomological amplitude) [Spec]. Let $R \in D^b(k\text{-mod})$ have cohomological amplitude contained in $[-1, 0]$. Over a field k , the universal coefficient spectral sequence collapses and yields a canonical isomorphism

$$\text{Ext}^1(R, k) \cong \text{Hom}(H^{-1}(R), k).$$

Therefore $\text{Ext}^1(R, k) = 0 \Rightarrow H^{-1}(R) = 0$. In the pipeline, the operational gate uses only the proven forward implication $\text{PH}_1 \Rightarrow \text{Ext}^1$ (Appendix C/D policy; logged assumptions in Appendix G), and does *not* rely on any global equivalence. A reverse implication $\text{Ext}^1 \Rightarrow \text{PH}_1$ may be audited *window-locally* only when additional, explicitly logged hypotheses identify $H^{-1}(\mathcal{R}(C_\tau F))$ with the relevant persistence obstruction on that window (e.g. via the Local Equiv/Overlap Gate prerequisites); this optional audit is outside the mandatory gate and must be marked as such in `run.yaml`.

Summary. The Betti integral equals clipped barcode mass for $\alpha = 1$ (Theorem E.1), hence $\text{PE}_i^{\leq \tau}$ is continuous, nondecreasing, and piecewise linear with breakpoints among births/deaths (Corollary E.2). Any affine-on-components constraint reduces to the finite event set (Proposition E.6); under the *o-minimal* definable policy (Appendix G), these events are a priori finite and computable (Remark E.7), so E_1 -level checks in the Overlap Gate become finite, auditable checklists. Deletion-type truncations make $\text{PE}_{i,\alpha}^{\leq \tau}$ nonincreasing and preserve 1-Lipschitz stability in interleaving distance (Appendix A). Windowed perturbation bounds (Proposition E.9) yield practical stability on bounded ranges. Finally, the `Lambda_len` audit (Appendix G) provides an independent length-spectrum certificate of the $\alpha = 1$ identity via trace and multiset checks (Corollary E.5); any reverse-bridge reasoning is optional, window-local, and must be explicitly logged as [Spec].

Appendix I. ε -Survival Lemma and Grid-to-Continuum [Proof/Spec] (reinforced)

Standing conventions. We work over a field k . All persistence modules lie in $\text{Pers}_k^{\text{ft}}$ (constructible; locally finite on bounded windows). Any use of filtered colimits follows the scope policy of Appendix A, Remark .67. Global conventions: Ext -tests are always against $k[0]$ (we write $\text{Ext}^1(\mathcal{R}(C_\tau F), k) = 0$, with C_τ understood up to f.q.i. on $\text{Ho}(\text{FiltCh}(k))$); the energy exponent satisfies $\alpha > 0$ (default $\alpha = 1$); type dashes (Type I–II / Type III / Type IV) are used uniformly. We write d_{int} for the interleaving metric, and in the 1-parameter constructible regime we identify it with the bottleneck metric via the usual isometry theorem (when this identification is invoked, it is explicitly stated below). Window policy, right-open conventions, coverage checks, δ -budgets, and gating are as in Appendix D. All survival/stability statements in the pipeline are applied *after* the mandatory order

$$\boxed{\text{for each } t \Rightarrow \mathbf{P}_t \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare in } \text{Pers}_k^{\text{ft}}},$$

and any additional window restriction is implemented by $\mathbf{W}_{\leq \tau_0}$ defined in §I.1.

Remark F.1 (Endpoint conventions). Intervals are taken half-open $[b, d)$ with $d \in \mathbb{R} \cup \{\infty\}$. Any consistent open/closed choice yields the same clipped lengths and event sets on bounded windows (cf. Appendix H, Remark E.3); all quantitative statements below are invariant under this choice.

I.1. Window clipping and nonexpansivity

Let $\text{Pers}_k^{\text{ft}}$ denote constructible persistence modules on (\mathbb{R}, \leq) . For $\tau \geq 0$, write $i_{\leq \tau} : [0, \tau] \hookrightarrow \mathbb{R}$ for the inclusion and define the *window clip* endofunctor

$$\mathbf{W}_{\leq \tau} := (i_{\leq \tau})_!^0 \circ i_{\leq \tau}^* : \text{Pers}_k^{\text{ft}} \longrightarrow \text{Pers}_k^{\text{ft}},$$

i.e. restriction to $[0, \tau]$ followed by extension by 0 outside $[0, \tau]$. On barcodes, $\mathbf{W}_{\leq \tau}$ corresponds to intersecting bars with $[0, \tau]$ and discarding empty intersections.

For a bar $I = [b, d)$, its clipped length on $[0, \tau_0]$ is

$$\ell_{[0, \tau_0]}(I) := \lambda(I \cap [0, \tau_0]) = \max\{0, \min\{d, \tau_0\} - \max\{b, 0\}\}, \quad \min\{\infty, \tau_0\} = \tau_0.$$

Remark F.2 (Shifts commute with clipping). For every $\varepsilon \geq 0$, there is a natural isomorphism

$$S^\varepsilon \circ \mathbf{W}_{\leq \tau} \cong \mathbf{W}_{\leq \tau} \circ S^\varepsilon.$$

Consequently, any ε -interleaving transports through $\mathbf{W}_{\leq \tau}$.

Lemma F.3 (Clipping is 1-Lipschitz). *For all $M, N \in \text{Pers}_k^{\text{ft}}$ and $\tau \geq 0$,*

$$d_{\text{int}}(\mathbf{W}_{\leq \tau} M, \mathbf{W}_{\leq \tau} N) \leq d_{\text{int}}(M, N).$$

Proof. If M, N are ε -interleaved, there are morphisms $f : M \rightarrow S^\varepsilon N$ and $g : N \rightarrow S^\varepsilon M$ satisfying the standard triangle conditions. Applying $\mathbf{W}_{\leq \tau}$ and using Remark F.2 yields morphisms $\mathbf{W}_{\leq \tau} M \rightarrow S^\varepsilon \mathbf{W}_{\leq \tau} N$ and $\mathbf{W}_{\leq \tau} N \rightarrow S^\varepsilon \mathbf{W}_{\leq \tau} M$ satisfying the same triangle conditions. Hence $\mathbf{W}_{\leq \tau} M, \mathbf{W}_{\leq \tau} N$ are ε -interleaved. Taking the infimum over ε gives the claim. \square

I.2. Survival under ε -interleavings

We state a sharp clipped-length survival estimate. When we refer to an ε -matching of bars, we use the 1D isometry theorem in the constructible regime to pass from $d_{\text{int}} \leq \varepsilon$ to a bottleneck-style endpoint matching.

Lemma F.4 (ε -survival; sharp two-sided clipped-length form). *Let $M, N \in \text{Pers}_k^{\text{ft}}$ satisfy $d_{\text{int}}(M, N) \leq \varepsilon$, and fix $\tau_0 > 0$. Assume we choose an ε -matching of bars (equivalently, an ε -bottleneck matching under the 1D isometry theorem). Then:*

1. *If a bar I of M is matched to a bar J of N , then*

$$\ell_{[0, \tau_0]}(J) \geq \max\{\ell_{[0, \tau_0]}(I) - 2\varepsilon, 0\}.$$

2. *If $\ell_{[0, \tau_0]}(I) > 2\varepsilon$, then its matched partner J satisfies $\ell_{[0, \tau_0]}(J) > 0$.*

3. (Multiplicity) *If at least r bars in M have $\ell_{[0, \tau_0]}(\cdot) > 2\varepsilon$ (counted with multiplicity), then $\mathbf{W}_{\leq \tau_0} N$ has at least r nonzero bars (with multiplicity).*

Proof. Under an ε -matching, matched endpoints satisfy $|b - b'| \leq \varepsilon$ and $|d - d'| \leq \varepsilon$ (with $d = \infty$ allowed). Clipping to $[0, \tau_0]$ changes the left truncation point $\max\{b, 0\}$ by at most ε , and changes the right truncation point $\min\{d, \tau_0\}$ by at most ε . Therefore the clipped length changes by at most 2ε , yielding (1). Items (2) and (3) follow immediately. \square

Remark F.5 (After-collapse composition). The bar-deletion reflector \mathbf{T}_τ is 1-Lipschitz for d_{int} (Appendix A). Therefore Lemma F.4 remains valid after pre/post-composition by \mathbf{T}_τ , and in particular it applies to the pipeline objects $\mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F)$ and $\mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F_h)$.

I.3. Grid-to-continuum transfer

Let F be filtered with degree- i persistence $\mathbf{P}_i(F)$, and let F_h be a discretization (grid approximation) such that

$$d_{\text{int}}(\mathbf{P}_i(F_h), \mathbf{P}_i(F)) \leq \varepsilon(h).$$

Fix a pipeline truncation scale $\tau > 0$ and a window $[0, \tau_0]$. Define the post-collapse/windowed modules

$$M_h := \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F_h), \quad M := \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F).$$

By nonexpansivity of \mathbf{T}_τ and $\mathbf{W}_{\leq \tau_0}$, $d_{\text{int}}(M_h, M) \leq \varepsilon(h)$.

Theorem F.6 (Grid-to-continuum survival). *Fix $\tau_0 > 0$, $r \in \mathbb{Z}_{\geq 1}$, and $\eta > 0$. If $M_h = \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F_h)$ has at least r bars of clipped length $\geq 2\varepsilon(h) + \eta$, then $M = \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F)$ has at least r nonzero bars, each of clipped length $\geq \eta$.*

Proof. Apply Lemma F.4 to M_h, M with $\varepsilon = \varepsilon(h)$. □

I.4. Budget-adjusted and V -metric variants [Spec]

We now incorporate (i) the δ -ledger/budget of Appendix G and (ii) an optional V -enriched (quantale) distance used in the suite.

Setup. Let d_V be a Lawvere/quantale distance on $\text{Pers}_k^{\text{ft}}$ such that both $\mathbf{W}_{\leq \tau}$ and \mathbf{T}_τ are 1-Lipschitz. Let $W = [0, \tau_0]$ be the active window. Let Δ_W denote the window-restricted budget aggregate obtained by combining the operation-level residuals operations[*].delta.total via the quantale operation \oplus recorded in run.yaml.quantale.op (Appendix G). (In the default $[0, \infty]_+$ quantale, \oplus is ordinary addition.) Define the *effective radius*

$$\varepsilon_{\text{eff}} := d_V(\mathbf{P}_i(F_h), \mathbf{P}_i(F)) \oplus \Delta_W.$$

Lemma F.7 (ε -survival, budgeted V -metric). *With notation as above, consider $M_h := \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F_h)$ and $M := \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F)$. Assume $d_V(M_h, M) \leq \varepsilon_{\text{eff}}$. Then for any bar I in M_h :*

1. (Two-sided, sharp) *If $\ell_{[0, \tau_0]}(I) > 2\varepsilon_{\text{eff}}$, then the matched bar in M is nonzero; moreover its clipped length is*

$$\geq \ell_{[0, \tau_0]}(I) \ominus 2\varepsilon_{\text{eff}},$$

where \ominus denotes the truncated subtraction in the default additive case (i.e. $\max\{x - 2\varepsilon_{\text{eff}}, 0\}$); equivalently, the loss is bounded by $2\varepsilon_{\text{eff}}$.

2. (One-sided improvement; optional [Spec]) *If the implementation enforces a one-sided control in which births are anchored and only the death-side may drift by $\leq \varepsilon_{\text{eff}}$, then the nonvanishing threshold improves to $\ell_{[0, \tau_0]}(I) > \varepsilon_{\text{eff}}$, and the residual margin degrades by at most ε_{eff} .*

Proof sketch. Nonexpansivity of $\mathbf{W}_{\leq \tau_0}$ and \mathbf{T}_τ in d_V yields $d_V(M_h, M) \leq \varepsilon_{\text{eff}}$. In the default additive regime, the two-sided survival bound follows by the same endpoint-motion argument as Lemma F.4, with ε replaced by ε_{eff} . The one-sided improvement is implementation-specific and follows if only one endpoint is permitted to drift. □

Corollary F.8 (Budgeted grid-to-continuum). *If at least r clipped bars in $M_h = \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F_h)$ have length $\geq 2\varepsilon_{\text{eff}} + \eta$, then $M = \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F)$ has at least r nonzero bars of clipped length $\geq \eta$. Under the one-sided hypothesis of Lemma F.7(2), replace 2 by 1.*

Remark F.9 (Tropical bins and aux-bars; persistence-side diagnostic). If a diagnostic bin width $\beta > 0$ is declared (Appendix G, `tropical.bins.width`), then an ε_{eff} -drift implies a histogram/bin-profile shift bounded by

$$q = \left\lceil \varepsilon_{\text{eff}} / \beta \right\rceil$$

bins (in the default additive regime). When this diagnostic is used, record `eps_cont_shift_bins = q` in `run.yaml`.

Remark F.10 (Logging (minimal contract)). Record in `run.yaml` (Appendix G): the metric name (`pipeline.metric` or `quantale.name/dV` identifier); the active window $W = [0, \tau_0]$ under `windows.domain`; `filtration_range`; the contributing δ -entries and their aggregation rule (`quantale.op`, `budget.sum_delta`, `layered_delta`); the computed ε_{eff} ; counts of bars above $2\varepsilon_{\text{eff}} + \eta$ (or $\varepsilon_{\text{eff}} + \eta$ in one-sided mode); and confirmation that `policy.after_collapse_only = true`.

Corollary F.11 (Betti-integral lower bound (link to Appendix H)). *If $M_h = \mathbf{W}_{\leq \tau_0} \mathbf{T}_\tau \mathbf{P}_i(F_h)$ contains r bars of clipped length $\geq 2\varepsilon_{\text{eff}} + \eta$, then*

$$\text{PE}_i^{\leq \tau_0}(\mathbf{T}_\tau \mathbf{P}_i(F)) \geq r \eta,$$

by Theorem E.1 applied to $\mathbf{T}_\tau \mathbf{P}_i(F)$ and Corollary F.8. Since \mathbf{T}_τ is deletion-type, $\text{PE}_i^{\leq \tau_0}(\mathbf{P}_i(F)) \geq \text{PE}_i^{\leq \tau_0}(\mathbf{T}_\tau \mathbf{P}_i(F))$, so the same lower bound holds a fortiori for $\mathbf{P}_i(F)$.

I.5. Variants, sharpness, and towers

- *Sharpness.* The constant 2 in Lemma F.4 is optimal: a bar of clipped length 2ε can be shifted by ε at both ends to vanish on $[0, \tau_0]$.
- *After-collapse pipeline.* Since \mathbf{T}_τ is 1-Lipschitz (Appendix A), composing with \mathbf{T}_τ does not change survival thresholds; all inequalities are preserved under the mandated order $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$.
- *Towers.* If $d_V(\mathbf{P}_i(F_n), \mathbf{P}_i(F_\infty)) \leq \varepsilon_n \rightarrow 0$, then any fixed positive margin detected at stage n propagates to the limit by Corollary F.8. This interfaces with the tower comparison diagnostics (μ , u) and stability bands (Appendix D, Appendix G).

I.6. Formalization stubs (Lean/Coq) [Spec]

```
-- Lean-style pseudocode (schematic)
namespace AK.I
open scoped Classical
noncomputable section

/-- Constructible persistence modules (schematic placeholder). -/
structure PersModule := (dummy : Unit)

/-- Window clip and truncation (schematic). -/
def W_le (τ : ℝ≥0) : PersModule → PersModule := sorry
def Tτ (τ : ℝ≥0) : PersModule → PersModule := sorry

/-- V-distance (quantale-style) and nonexpansivity axioms. -/
def dV : PersModule → PersModule → ℝ≥0 := sorry
axiom W_le_non-expansive : ∀ τ M N, dV (W_le τ M) (W_le τ N) ≤ dV M N
axiom Tτ_non-expansive : ∀ τ M N, dV (Tτ τ M) (Tτ τ N) ≤ dV M N
```

```

/-- Bars and clipped lengths (schematic). -/
constant Bar : Type
constant BarOf : PersModule → Type
def clipLen_on (τ0 : ℝ≥0) : Bar → ℝ≥0 := sorry

theorem eps_survival_two_sided
  (τ0 : ℝ≥0) (εeff : ℝ≥0) {Mh M : PersModule}
  (h : dV Mh M ≤ εeff)
  (I : BarOf Mh) (hI : clipLen_on τ0 (Classical.choice Bar, by trivial) > 2*εeff) :
  ∃ J : BarOf M, True := by
  -- Push h through W_le and Tτ (non-expansive),
  -- then use endpoint drift ≤ εeff at both ends.
  admit

theorem eps_survival_one_sided -- optional birth-anchored mode
  (τ0 : ℝ≥0) (εeff : ℝ≥0) {Mh M : PersModule}
  (h : dV Mh M ≤ εeff) (births_anchored : True)
  (I : BarOf Mh) (hI : clipLen_on τ0 (Classical.choice Bar, by trivial) > εeff) :
  ∃ J : BarOf M, True := by
  admit

end AK.I

```

I.7. Summary

Clipping $\mathbf{W}_{\leq \tau}$ is restriction to $[0, \tau]$ followed by 0-extension; it preserves constructibility and is 1-Lipschitz (Lemma F.3). Under an ε -interleaving (equivalently, an ε -matching in the 1D constructible regime), clipped lengths degrade by at most 2ε ; thus any bar with clip-length $> 2\varepsilon$ survives as a nonzero bar on the same window (Lemma F.4). Applying this after the mandated order $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$ yields the grid-to-continuum survival transfer (Theorem F.6). In the budgeted V -metric setting, replace ε by the *effective* radius

$$\varepsilon_{\text{eff}} = d_V(\mathbf{P}_i(F_h), \mathbf{P}_i(F)) \oplus \Delta_W,$$

so the survival threshold becomes $> 2\varepsilon_{\text{eff}}$ (or $> \varepsilon_{\text{eff}}$ under the optional one-sided, birth-anchored mode). These statements commute with the after-collapse policy (\mathbf{T}_τ is 1-Lipschitz), tie into Betti-integral lower bounds (Appendix H; Corollary F.11), and are logged via the δ -ledger and window fields in Appendix G. No further supplementation is required for operational deployment or third-party audit.

Appendix J. Calculus of μ, u [Proof + Stability Bands + Window Pasting] (reinforced)

Standing conventions. We work over a field k . All persistence modules lie in the one-parameter constructible persistence category $\text{Pers}_k^{\text{cons}}$ (locally finite on bounded windows). Filtered colimits, when used, are computed in the functor category $[\mathbb{R}, \text{Vect}_k]$ under the scope policy of Appendix A, Remark .67, and are returned to the constructible range only when the policy conditions guarantee constructibility. The bar-deletion reflector $\mathbf{T}_\tau \dashv \iota_\tau$ is exact and 1-Lipschitz (Appendix A, Theorem .64 and Proposition .70). Global conventions: Ext^1 -tests are always against $k[0]$; the energy exponent $\alpha > 0$ (default $\alpha = 1$). All window

policies and overlap conventions follow Appendix G (MECE accounting with right-open primitives; overlap lists and finite-check manifests are mandatory). All (μ, u) diagnostics are computed *after* applying \mathbf{T}_τ (B-side single-layer policy), and always under the same τ and window record used by B-Gate⁺ and the δ -ledger (Appendix G).

J.0. Setup: towers, comparison maps, and generic fiber dimension

Towers. Fix $i \in \mathbb{Z}$. Let $F = (F_n)_{n \in I}$ be a directed system (“tower”) of filtered objects such that $\mathbf{P}_i(F_n) \in \text{Pers}_k^{\text{cons}}$ for all n . Let F_∞ denote a chosen colimit object with cocone maps $F_n \rightarrow F_\infty$. For $\tau \geq 0$ consider the *comparison map* in $[\mathbb{R}, \text{Vect}_k]$,

$$\phi_{i,\tau}(F) : \varinjlim_n \mathbf{T}_\tau(\mathbf{P}_i(F_n)) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(F_\infty)). \quad (\text{G.1})$$

Whenever $\phi_{i,\tau}(F)$ is asserted to lie in $\text{Pers}_k^{\text{cons}}$, this is by the scope policy hypotheses of Appendix A and the window/definability policies of Appendix G.

Generic fiber dimension. For a constructible module $M \in \text{Pers}_k^{\text{cons}}$, there exists t_0 such that $\dim_k M(t)$ stabilizes for all $t \geq t_0$. Define the *generic fiber dimension*

$$\dim_k^{\text{gen}}(M) := \dim_k M(t) \text{ for any (hence all) sufficiently large } t,$$

equivalently the stabilized right-tail dimension. In barcode terms, $\dim_k^{\text{gen}}(M)$ equals the total multiplicity of infinite bars $[b, \infty)$ (for any sufficiently large tail parameter), and after applying \mathbf{T}_τ it is computed on the post-deletion barcode (Appendix D, Remark A.2).

Definition of μ, u . Define

$$\mu_{i,\tau}(F) := \dim_k^{\text{gen}} \ker \phi_{i,\tau}(F), \quad u_{i,\tau}(F) := \dim_k^{\text{gen}} \text{coker } \phi_{i,\tau}(F), \quad (\text{G.2})$$

with \ker, coker taken in $[\mathbb{R}, \text{Vect}_k]$ and then interpreted in the constructible regime on the stabilized tail when permitted by policy.² We write $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ for the verdict $(\mu_{i,\tau}(F), u_{i,\tau}(F)) = (0, 0)$ and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \neq (0, 0)$ otherwise (Type IV witness at (i, τ)). We also write the totals

$$\mu_{\text{Collapse},\tau}(F) := \sum_i \mu_{i,\tau}(F), \quad u_{\text{Collapse},\tau}(F) := \sum_i u_{i,\tau}(F). \quad (\text{G.3})$$

which are finite when F is bounded in homological degrees (constructible range assumption). All quantities depend on τ ; *no global monotonicity in τ is asserted*.

J.1. Functoriality, isomorphism invariance, and cofinality

Definition G.1 (Morphisms of towers). A morphism $u : F \rightarrow G$ consists of maps $u_n : F_n \rightarrow G_n$ commuting with the structure maps and inducing a canonical map on colimits $u_\infty : F_\infty \rightarrow G_\infty$.

²Under constructibility, kernels/cokernels decompose into finite direct sums of interval modules on bounded windows, and \dim_k^{gen} equals the stabilized tail dimension, hence is well-defined and integer-valued. The diagnostics (μ, u) are the tower-level “infinite-bar” defect measures after deletion, and are designed to detect Type IV (invisible) failures.

Lemma G.2 (Naturality of comparison maps). *Under the scope policy, each tower morphism $u : F \rightarrow G$ induces a commutative square*

$$\begin{array}{ccc} \varinjlim_n \mathbf{T}_\tau \mathbf{P}_i(F_n) & \xrightarrow{\phi_{i,\tau}(F)} & \mathbf{T}_\tau \mathbf{P}_i(F_\infty) \\ \downarrow \varinjlim_n \mathbf{T}_\tau \mathbf{P}_i(u_n) & & \downarrow \mathbf{T}_\tau \mathbf{P}_i(u_\infty) \\ \varinjlim_n \mathbf{T}_\tau \mathbf{P}_i(G_n) & \xrightarrow{\phi_{i,\tau}(G)} & \mathbf{T}_\tau \mathbf{P}_i(G_\infty), \end{array}$$

natural in both i and τ , and compatible with composition of tower morphisms.

Proof. Apply \mathbf{P}_i , then \mathbf{T}_τ , and then filtered colimits in $[\mathbb{R}, \text{Vect}_k]$. Exactness of \mathbf{T}_τ and functoriality of colimits yield the square and compatibility with composition. \square

Remark G.3 (Invariance under isomorphism of towers). If $u : F \rightarrow G$ is a tower isomorphism (levelwise and on apex), then $\phi_{i,\tau}(F)$ and $\phi_{i,\tau}(G)$ are isomorphic as morphisms; hence $\mu_{i,\tau}(F) = \mu_{i,\tau}(G)$ and $u_{i,\tau}(F) = u_{i,\tau}(G)$.

Definition G.4 (Cofinal subindexing). Let I be the directed index category for F . A full subcategory $J \subset I$ is cofinal if for every $i \in I$ there exists $j \in J$ with a morphism $i \rightarrow j$. The restricted tower $F|_J$ has the same colimit as F in $[\mathbb{R}, \text{Vect}_k]$.

Theorem G.5 (Cofinal invariance). *Let $J \subset I$ be cofinal. Then for all $\tau \geq 0$,*

$$\mu_{i,\tau}(F|_J) = \mu_{i,\tau}(F), \quad u_{i,\tau}(F|_J) = u_{i,\tau}(F).$$

Hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$ agree as well.

Proof. Cofinal restriction does not change colimits in $[\mathbb{R}, \text{Vect}_k]$, hence does not change the source, target, or definition of $\phi_{i,\tau}(F)$. Kernels/cokernels and stabilized tail dimensions therefore agree. \square

J.2. Linear-algebra calculus for stabilized kernels/cokernels (P6)

A map-level notation (post- \mathbf{T}_τ). For a morphism $f : M \rightarrow N$ in the constructible regime (or in $[\mathbb{R}, \text{Vect}_k]$ under stabilization), define

$$\mu^{\text{gen}}(f) := \dim_k^{\text{gen}} \ker(f), \quad u^{\text{gen}}(f) := \dim_k^{\text{gen}} \text{coker}(f).$$

With this notation, $\mu_{i,\tau}(F) = \mu^{\text{gen}}(\phi_{i,\tau}(F))$ and $u_{i,\tau}(F) = u^{\text{gen}}(\phi_{i,\tau}(F))$.

Theorem G.6 (Subadditivity under composition (P6)). *Let $f : M \rightarrow N$ and $g : N \rightarrow P$ be composable morphisms in the stabilized constructible regime (in particular, after the mandated \mathbf{T}_τ step). Then*

$$\mu^{\text{gen}}(g \circ f) \leq \mu^{\text{gen}}(f) + \mu^{\text{gen}}(g), \quad u^{\text{gen}}(g \circ f) \leq u^{\text{gen}}(f) + u^{\text{gen}}(g).$$

Proof. Choose $t \gg 0$ in a common stabilization tail for all modules involved, so that $\dim_k(\ker(\cdot))(t)$ and $\dim_k(\text{coker}(\cdot))(t)$ equal the generic dimensions. Evaluating at t yields finite-dimensional linear maps of k -spaces $f_t : M_t \rightarrow N_t$ and $g_t : N_t \rightarrow P_t$. For linear maps $A : U \rightarrow V$, $B : V \rightarrow W$, one has

$$\dim \ker(B \circ A) \leq \dim \ker(A) + \dim \ker(B), \quad \dim \text{coker}(B \circ A) \leq \dim \text{coker}(A) + \dim \text{coker}(B),$$

the first by the exact sequence $0 \rightarrow \ker A \rightarrow \ker(BA) \rightarrow \ker B \cap \text{Im} A \rightarrow 0$, the second by duality or rank inequalities. Applying these to $A = f_t$, $B = g_t$, and reading the stabilized values gives the claim. \square

Remark G.7 (Metric-free; valid in V -enriched regimes). Theorem G.6 uses only tail-stabilized linear algebra; it is independent of any metric, and therefore remains valid verbatim in the V -enriched setting of Appendices E–F.

Remark G.8 (How P6 is used in the pipeline). In the operational stack, P6 is applied to compositions formed by post-processing maps that are required to be non-expansive and deletion-type (Appendix G), e.g. $\mathbf{W}_{\leq \tau_0}$, \mathbf{T}_τ , and permitted continuation maps. Because (μ, u) are computed on the B-side after \mathbf{T}_τ , such post-processing never creates “budget-escaping” decreases in the diagnostic slack beyond what is logged in the δ -ledger.

J.3. Additivity under finite direct sums

Proposition G.9 (Direct-sum additivity). *Let $F = F^{(1)} \oplus F^{(2)}$ be the levelwise direct sum of towers (same index category), and similarly for the colimit. Then for every $\tau \geq 0$,*

$$\mu_{i,\tau}(F) = \mu_{i,\tau}(F^{(1)}) + \mu_{i,\tau}(F^{(2)}), \quad u_{i,\tau}(F) = u_{i,\tau}(F^{(1)}) + u_{i,\tau}(F^{(2)}).$$

Hence $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0) / (\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0)$ is additive under finite direct sums as well.

Proof. \mathbf{P}_i and \mathbf{T}_τ preserve finite direct sums; filtered colimits commute with finite direct sums in $[\mathbb{R}, \text{Vect}_k]$. Thus $\phi_{i,\tau}(F)$ is block-diagonal with blocks $\phi_{i,\tau}(F^{(1)})$, $\phi_{i,\tau}(F^{(2)})$; kernels/cokernels and stabilized tail dimensions add. \square

J.4. Vanishing criteria (when the tower is “well-behaved”)

Proposition G.10 (Sufficient conditions for $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$). *Assume one of the sufficient conditions of Appendix D, §D.3 holds for the tower F at the monitored (i, τ) (e.g. commutation of \mathbf{T}_τ with the relevant colimit in the scoped sense (S1), no τ -accumulation plus constructible return (S2), or a \mathbf{T}_τ -Cauchy tower with compatible cocone (S3)). Then $\phi_{i,\tau}(F)$ is an isomorphism on the stabilized tail, hence*

$$\mu_{i,\tau}(F) = u_{i,\tau}(F) = 0.$$

Proof. Under any of (S1)–(S3) (Appendix D), the comparison map is an isomorphism after applying the mandated \mathbf{T}_τ , at least on the stabilized right tail where \dim_k^{gen} is computed. Thus $\ker \phi_{i,\tau}(F)$ and $\text{coker } \phi_{i,\tau}(F)$ have zero generic dimension. \square

Remark G.11 (Interpretation: Type IV detector). By design, (μ, u) detect “invisible” defects that may not appear in window-local E_1 -style checks. Operationally: $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ is the normal state; $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0)$ is a Type IV witness at the monitored (i, τ) , and must be logged and surfaced by the Auditor layer (Appendix G).

J.5. τ -sweeps and stability bands

Definition G.12 (τ -sweep and stability band). Fix a window (MECE/right-open accounting) and a degree i . A τ -sweep is a finite increasing list $\{\tau_\ell\}_{\ell=0}^L \subset (0, \infty)$ recorded in `run.yaml`. A contiguous subarray $\{\tau_a, \dots, \tau_b\}$ is a *stability band* for F in degree i if

$$\mu_{i,\tau_\ell}(F) = u_{i,\tau_\ell}(F) = 0 \quad \text{for all } \ell \in \{a, \dots, b\},$$

and the verdict persists under refinement of the sweep (i.e. inserting additional τ -values into $[\tau_a, \tau_b]$ does not create $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq } (0, 0)$ there).

Proposition G.13 (Robust detection under refinement). *Assume the definable/no-accumulation window policy of Appendix G and that one of Appendix D’s sufficient hypotheses (S1)–(S3) holds on an interval of τ -values around τ_0 . Then there exists an open neighborhood U of τ_0 such that $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ holds for all $\tau \in U$. A sufficiently fine τ -sweep detects U as a stability band and remains stable under refinement.*

Proof. Under (S1)–(S3) on a neighborhood, $\phi_{i,\tau}(F)$ is an isomorphism on the stabilized tail throughout that neighborhood, hence $\mu = u = 0$ there. Refining the sweep cannot change the evaluated values, so the detected band is stable. \square

J.6. Piecewise constancy off a finite critical set (bounded τ -ranges)

Proposition G.14 (Finite critical set and piecewise constancy). *Fix a tower F , degree i , and a bounded interval $[a, b] \subset (0, \infty)$. Assume the definable/no- τ -accumulation policy of Appendix G for the relevant bar-data after \mathbf{T}_τ . Then there exists a finite set $S \subset [a, b]$ such that $\mu_{i,\tau}(F)$ and $u_{i,\tau}(F)$ are locally constant on each connected component of $[a, b] \setminus S$.*

Proof sketch. Under definability (or an explicit no-accumulation contract) on bounded τ -ranges, the set of τ -values at which the post- \mathbf{T}_τ barcode data can change is finite on $[a, b]$. The stabilized tail linear-algebra type of $\phi_{i,\tau}(F)$ (hence the generic dimensions of its kernel and cokernel) can change only at those critical τ -values. Therefore $\mu_{i,\tau}$ and $u_{i,\tau}$ are constant between successive critical values. \square

Corollary G.15 (Band openness). *If $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ holds at some $\tau_0 \in (a, b)$, then it holds on an open neighborhood of τ_0 inside (a, b) . Hence stability bands are unions of open intervals intersected with the sweep list.*

J.7. Window pasting via Restart and Summability [Spec/Policy]

Definition G.16 (Per-window budget and safety gap). Let $\{W_k\}$ be a MECE accounting of a domain, with right-open primitives and an explicit overlap list (Appendix G). On each window W_k , let $\tau_k > 0$ be the selected collapse threshold (typically chosen inside a stability band when available). Define the *window budget* (degree-wise) by aggregating the δ -ledger entries recorded for that window:

$$\Sigma\delta_k(i) := \bigoplus_{\text{op} \in \text{Ops}(W_k)} \delta_{\text{op}}(i, \tau_k),$$

where \oplus is the quantale/budget aggregator declared in run.yaml (default: ordinary addition). Let $\text{gap}_{\tau_k}(i) > 0$ be the configured slack margin required by B-Gate⁺ on W_k in degree i .

Lemma G.17 (Restart inequality (policy form)). *Assume the transition from W_k to W_{k+1} is realized by a finite composition of permitted deletion-type steps and ε -continuations, all measured after \mathbf{T}_{τ_k} and recorded in the manifest. If B-Gate⁺ passes on W_k with $\text{gap}_{\tau_k}(i) > \Sigma\delta_k(i)$, then there exists $\kappa \in (0, 1]$, determined by the admissible step class and the declared τ -adaptation policy, such that*

$$\text{gap}_{\tau_{k+1}}(i) \geq \kappa \left(\text{gap}_{\tau_k}(i) \ominus \Sigma\delta_k(i) \right),$$

where \ominus denotes truncated subtraction in the default additive budget regime (i.e. $\max\{x - \Sigma\delta, 0\}$).

Remark G.18 (Status). Lemma G.17 is a policy contract: the implementation must either (i) supply a proof for the chosen step class and metric model, or (ii) treat κ and the inequality as an explicit assumption logged under assumptions.restart_inequality (Appendix G).

Definition G.19 (Summability). A run satisfies *Summability* (on monitored degrees $i \in I$) if

$$\sum_k \Sigma \delta_k(i) < \infty \quad \text{for all } i \in I,$$

with the series interpretation determined by \oplus (default: ordinary series in $[0, \infty]$). A sufficient pattern is geometric decay of window scales (e.g. $\tau_k = \tau_0 \rho^k$, $\rho \in (0, 1)$) with uniformly bounded per-window operation counts and bounded per-operation δ entries (Appendix G).

Theorem G.20 (Pasting windowed certificates (policy theorem)). *Let $\{W_k\}$ be a MECE accounting of a domain with explicit overlap lists. Suppose that on each W_k , $B\text{-Gate}^+$ passes with $\text{gap}_{\tau_k}(i) > \Sigma \delta_k(i)$ for all monitored degrees $i \in I$, that the Restart inequality (Lemma G.17) holds at every transition, and that Summability (Definition G.19) is satisfied. Then the concatenation of per-window certificates yields a global certificate on $\bigcup_k W_k$ for all monitored degrees $i \in I$, with all budgets and overlaps auditable from the manifest (Appendix G).*

J.8. Countable covers and finite Čech depth (T-Countable-Cover)

Theorem G.21 (Countable cover; bounded overlap depth implies finite Čech depth). *Let $\mathcal{W} = \{W_j\}_{j \in \mathbb{N}}$ be a countable right-open cover of a bounded interval $U \subset \mathbb{R}$. Assume a bounded overlap multiplicity certificate:*

$$m := \sup_{t \in U} \#\{j \mid t \in W_j\} < \infty,$$

either (i) provided explicitly as a checked bound, or (ii) derived from definability as in Theorem G.23. Then:

1. *The Čech nerve of \mathcal{W} truncates in degree $m - 1$.*
2. *Overlap Glue therefore reduces to finitely many overlap checks on each compact subinterval of U (bounded by m and the logged overlap lists), consistent with the finite-check doctrine of Appendices G–H.*

Proof. If $m < \infty$, then any p -fold intersection for $p > m$ is empty, so the Čech nerve has no simplices in degrees $\geq m$, hence truncates in degree $m - 1$. Finite-check reduction follows because only finitely many overlap patterns up to depth m can appear on compact subintervals once overlap lists are logged. \square

Specification (T-Countable-Cover). *Input:* a countable right-open cover \mathcal{W} of bounded U ; either an explicit bound m or definability metadata. *Checks:* (i) compute/verify m , (ii) record Čech depth $m - 1$, (iii) store finite overlap lists used by Overlap Glue. *Accept iff* these fields appear in `run.yaml` and are referenced by the Overlap Glue routine.

J.9. Summability contract (T-Delta-Sum-Converges)

Proposition G.22 (Summable δ -ledger via comparison tests). *Let $\{W_k\}_{k \in \mathbb{N}}$ be a (logged) countable window accounting on a bounded U . Suppose the per-window budgets satisfy either*

$$\Sigma \delta_k(i) \leq C \rho^k \quad (C > 0, \rho \in (0, 1)), \quad \text{or} \quad \Sigma \delta_k(i) \leq C \cdot \psi(k) \quad \text{with} \quad \sum_k \psi(k) < \infty.$$

Then $\sum_k \Sigma \delta_k(i) < \infty$ for all monitored degrees i , i.e. Summability holds.

Proof. Immediate by comparison with a convergent geometric series or the reference series $\sum_k \psi(k)$. \square

Specification (T-Delta-Sum-Converges). *Input:* recorded sequence $\{\Sigma\delta_k(i)\}$ and a proof obligation (geometric bound or reference-series bound). *Check:* compute partial sums; certify a finite bound B_i for each i . *Accept iff* $\sup_n \sum_{k \leq n} \Sigma\delta_k(i) \leq B_i < \infty$ is logged and referenced by B-Gate⁺ in run.yaml.

J.10. Definable Čech finiteness (o-minimal)

Theorem G.23 (Definable Čech finiteness). *Let $\mathcal{W} = \{W_j\}_{j \in J}$ be a right-open cover of a bounded interval U such that endpoints and overlaps are definable in an o-minimal structure (Appendix G). Then there exists a finite subcover, the overlap multiplicity m is finite, and the Čech nerve truncates in degree $m - 1$.*

Proof. In an o-minimal structure, definable subsets of \mathbb{R} are finite unions of points and intervals. Definability of endpoints/overlaps implies only finitely many combinatorial overlap types occur on bounded U , hence a finite subcover exists and multiplicity is uniformly bounded. Čech truncation follows from the finite bound m . \square

J.11. Mandatory after-collapse order and coherence contract

All diagnostics $\phi_{i,\tau}$, $\mu_{i,\tau}$, and $u_{i,\tau}$ are computed *after* applying \mathbf{T}_τ (B-side single layer), and *with the same window and the same τ* as used by B-Gate⁺ and the δ -ledger (Appendix G). This mandatory order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare}}$$

ensures that any subsequent 1-Lipschitz post-processing ($\mathbf{W}_{\leq \tau_0}$, continuation maps, or allowed V -non-expansive functors) does not invalidate budgets, does not create unlogged amplification, and keeps the Auditor/reproducibility manifest complete.

J.12. Minimal API sketch (pseudocode)

Compute (mu, nu) at scale tau for tower F and degree i (after T_tau)

def compute_mu_nu(F, i, tau):

 Ms = [T_tau(P_i(F_n), tau) for F_n in F.levels]

 M_inf = T_tau(P_i(F.apex), tau)

 colim_M = colim(Ms) # scoped colimit in [R, Vect_k]

 phi = comparison(colim_M, M_inf)

 mu = generic_tail_dim(kernel(phi)) # stabilized dimension t >> 0

 nu = generic_tail_dim(cokernel(phi))

 return mu, nu

Detect stability bands on a finite tau sweep; refinement-stable if piecewise-constant policy holds

def detect_stability_bands(F, i, tau_sweep):

 vals = [compute_mu_nu(F, i, tau) for tau in tau_sweep]

 zeros = [idx for idx, (mu, nu) in enumerate(vals) if (mu, nu) == (0, 0)]

 return maximal_contiguous_subarrays(zeros)

Summability check and cover depth contract

def summable_budget(deltas): # T-Delta-Sum-Converges

 return (sum(deltas) < +infty)

def cech_depth_bound(cover): # T-Countable-Cover

```

m = max_overlap_multiplicity(cover)    # must be certified/derived
return m-1

```

Summary. Within the constructible regime and the filtered-colimit scope of Appendix A, the tower comparison map $\phi_{i,\tau}(F)$ is natural under tower morphisms (Lemma G.2) and invariant under cofinal restriction (Theorem G.5). The diagnostics $(\mu_{i,\tau}(F), u_{i,\tau}(F))$ are defined as stabilized (generic tail) dimensions of \ker/coker of $\phi_{i,\tau}(F)$ after \mathbf{T}_τ ; $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0,0)$ is the designated Type IV witness. A core calculus principle (P6) is the subadditivity of stabilized kernel/cokernel dimensions under composition (Theorem G.6), independent of any metric. Additivity holds under finite direct sums (Proposition G.9). On bounded τ -ranges, under the definable/no-accumulation policy, $(\mu_{i,\tau}, u_{i,\tau})$ are piecewise constant off a finite critical set (Proposition G.14), enabling robust stability-band detection (Definition G.12, Proposition G.13). Window pasting is governed by the δ -ledger budgets, a Restart inequality, and Summability (Definitions G.16, G.19; Theorem G.20), with countable-cover overlap glue reduced to finite Čech depth once bounded multiplicity is certified (Theorem G.21) or derived via definability (Theorem G.23). All comparisons, budgets, and logs are made after collapse on the B-side single layer, preserving non-expansive bounds and reproducible audits (Appendix G).

Appendix K. Idempotent (Co)Monads for Collapse (up to f.q.i.) [Spec + Soft-Commuting + AWFS/2-Cell Budget] (reinforced)

Standing conventions. We work over a field k . All persistence modules are taken in the one-parameter constructible persistence category $\text{Pers}_k^{\text{cons}}$ (locally finite on bounded windows), and all equalities asserted at the persistence layer are equalities in $\text{Pers}_k^{\text{cons}}$. Filtered (co)limits, when invoked, are computed in the functor category $[\mathbb{R}, \text{Vect}_k]$ under the scope policy of Appendix A, Remark .67; when stated, the result is returned to the constructible range only under the logged hypotheses of Appendix A and the window/definability policies of Appendix G. The bar-deletion reflector $\mathbf{T}_\tau \dashv \iota_\tau$ onto the τ -local orthogonal subcategory $(E_\tau)^\perp \subset \text{Pers}_k^{\text{cons}}$ is exact and 1-Lipschitz (Appendix A, Theorem .64, Proposition .70). Global conventions: Ext^1 is always against $k[0]$; the energy exponent satisfies $\alpha > 0$ (default $\alpha = 1$); windows are MECE and right-open with overlap lists logged (Appendix G). Distances are measured by the interleaving metric d_{int} (which agrees with bottleneck distance on barcodes in the 1D constructible setting). Identifications “up to f.q.i.” occur in $\text{Ho}(\text{FiltCh}(k))$; they are preserved by \mathbf{P}_i whenever \mathbf{P}_i is applied. All soft-commuting and 2-cell budget claims are evaluated *after* the mandated order $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$ (B-side single layer), and must be measured on the same window/ τ as the gate and δ -ledger (Appendix G).

K.1. Persistence layer: the idempotent monad $\mathbf{M}_\tau = \iota_\tau \mathbf{T}_\tau$

Let $\iota_\tau : (E_\tau)^\perp \hookrightarrow \text{Pers}_k^{\text{cons}}$ be the fully faithful inclusion and $\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \rightarrow (E_\tau)^\perp$ its left adjoint (Appendix A). Set

$$\mathbf{M}_\tau := \iota_\tau \circ \mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \longrightarrow \text{Pers}_k^{\text{cons}}.$$

Theorem H.1 (Idempotent monad). *With unit $\eta : \text{Id} \Rightarrow \mathbf{M}_\tau$ and multiplication*

$$\mu : \mathbf{M}_\tau^2 = \iota_\tau \mathbf{T}_\tau \iota_\tau \mathbf{T}_\tau \xrightarrow{\iota_\tau \varepsilon \mathbf{T}_\tau} \iota_\tau \mathbf{T}_\tau = \mathbf{M}_\tau,$$

induced by the counit $\varepsilon : \mathbf{T}_\tau \iota_\tau \Rightarrow \text{Id}$ on $(E_\tau)^\perp$, the triple $(\mathbf{M}_\tau, \eta, \mu)$ is a monad and μ is a natural isomorphism (idempotence).

Proof. Any adjunction $\mathbf{T}_\tau \dashv \iota_\tau$ yields a monad $\iota_\tau \mathbf{T}_\tau$ with unit given by the adjunction unit and multiplication induced by the counit. Because ι_τ is fully faithful, the counit $\varepsilon : \mathbf{T}_\tau \iota_\tau \Rightarrow \text{Id}$ is a natural isomorphism. Hence $\mu = \iota_\tau \varepsilon \mathbf{T}_\tau$ is a natural isomorphism, i.e. \mathbf{M}_τ is idempotent. \square

Proposition H.2 (Exact and non-expansive). \mathbf{M}_τ is exact and 1-Lipschitz:

$$d_{\text{int}}(\mathbf{M}_\tau M, \mathbf{M}_\tau N) \leq d_{\text{int}}(M, N) \quad \text{for all } M, N \in \text{Pers}_k^{\text{cons}}.$$

Proof. Exactness and 1-Lipschitzness of \mathbf{T}_τ are given in Appendix A. The inclusion ι_τ is fully faithful and exact on the essential image, and it does not increase interleaving distance. Therefore the composite $\mathbf{M}_\tau = \iota_\tau \mathbf{T}_\tau$ is exact and 1-Lipschitz. \square

Remark H.3 (Fixed points and algebras). The Eilenberg–Moore category of \mathbf{M}_τ -algebras identifies with $(\mathbf{E}_\tau)^\perp$ via ι_τ . Equivalently, M is a fixed point (η_M an isomorphism) iff $M \in (\mathbf{E}_\tau)^\perp$. Thus \mathbf{M}_τ is the persistence-layer τ -collapse, idempotent on its essential image.

Example H.4 (Length threshold in 1D). In the 1D constructible (barcode) case, \mathbf{T}_τ deletes all interval summands of length $< \tau$ (Appendix A). Hence \mathbf{M}_τ is the canonical τ -deleted representative, and $\mathbf{M}_\tau^2 \simeq \mathbf{M}_\tau$.

K.2. Filtered layer: implementable idempotent comonad up to f.q.i. [Spec]

Spec context. This subsection is *[Spec]*: it records the intended filtered-layer mechanism consistent with the proven persistence-layer reflector. No claim here upgrades to a strict model-level statement unless explicitly discharged by an implementation proof and logged as such.

[Spec] There exists a coreflective “implementable” subcategory

$$\text{Ho}(\text{FiltCh}(k))_\tau^{\text{comb}} \subset \text{Ho}(\text{FiltCh}(k))$$

and a fully faithful inclusion $\iota : \text{Ho}(\text{FiltCh}(k))_\tau^{\text{comb}} \hookrightarrow \text{Ho}(\text{FiltCh}(k))$ with right adjoint (coreflector) C_τ^{comb} (natural up to f.q.i.). Define the endofunctor

$$\mathbf{G}_\tau := \iota \circ C_\tau^{\text{comb}} : \text{Ho}(\text{FiltCh}(k)) \longrightarrow \text{Ho}(\text{FiltCh}(k)).$$

Theorem H.5 ([Spec] Idempotent comonad up to f.q.i.). *With counit $\varepsilon : \mathbf{G}_\tau \Rightarrow \text{Id}$ (the coreflection counit) and comultiplication*

$$\delta : \mathbf{G}_\tau \xrightarrow{\iota \eta C_\tau^{\text{comb}}} \mathbf{G}_\tau^2,$$

($\mathbf{G}_\tau, \varepsilon, \delta$) is a comonad in $\text{Ho}(\text{FiltCh}(k))$; moreover δ is a natural isomorphism (idempotence). All statements are invariant under filtered quasi-isomorphism.

Proof sketch. Any coreflection $\iota \dashv C_\tau^{\text{comb}}$ yields a comonad $\iota C_\tau^{\text{comb}}$, with comultiplication induced by the adjunction unit. Because ι is fully faithful, the unit is an isomorphism on the essential image, which implies idempotence. All constructions occur in $\text{Ho}(\text{FiltCh}(k))$, hence are f.q.i.-invariant. \square

Proposition H.6 ([Spec] Compatibility with persistence; nonexpansivity after \mathbf{P}_i). *Naturally in i and F ,*

$$\mathbf{P}_i(\mathbf{G}_\tau F) \cong \mathbf{M}_\tau(\mathbf{P}_i F).$$

Moreover, for all F, G and all i ,

$$d_{\text{int}}(\mathbf{P}_i(\mathbf{G}_\tau F), \mathbf{P}_i(\mathbf{G}_\tau G)) \leq d_{\text{int}}(\mathbf{P}_i F, \mathbf{P}_i G).$$

Proof sketch. The first claim is the defining compatibility requirement for the implementable coreflector: it must realize the same post- τ deletion as \mathbf{T}_τ after applying \mathbf{P}_i . The second follows from 1-Lipschitzness of \mathbf{M}_τ (Proposition H.2) and functoriality of \mathbf{P}_i . \square

Remark H.7 (Scope). The comonad is asserted only on the implementable, f.q.i. range. This suffices for algorithms and stability accounting because all measured constraints are reduced to persistence-layer checks (after $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$) and recorded in the manifest (Appendix G).

K.3. Multiple reflectors: nesting, strict commutation, and soft-commuting policy

General torsion reflectors. Let E_A, E_B be hereditary Serre subcategories of $\text{Pers}_k^{\text{cons}}$ with exact reflectors T_A, T_B . Let $E_{A \vee B}$ denote the Serre join.

Proposition H.8 ([Spec] Nested torsions \Rightarrow strict order independence). *If $E_A \subseteq E_B$ or $E_B \subseteq E_A$, then*

$$T_A \circ T_B = T_B \circ T_A = T_{A \vee B}.$$

In particular, for deletion thresholds,

$$\mathbf{T}_\tau \circ \mathbf{T}_\sigma = \mathbf{T}_{\max\{\tau, \sigma\}} \quad \text{and hence} \quad \mathbf{M}_\tau \circ \mathbf{M}_\sigma = \mathbf{M}_{\max\{\tau, \sigma\}}.$$

Proof sketch. For nested torsion theories, applying the larger reflector already enforces orthogonality to the smaller torsion class, yielding order independence. The threshold identity follows because “delete bars of length $< \tau$ ” composed with “delete bars of length $< \sigma$ ” equals deleting bars of length $< \max\{\tau, \sigma\}$. \square

Definition H.9 (Commutation defect). For $M \in \text{Pers}_k^{\text{cons}}$, define the *commutation defect*

$$\Delta_{\text{comm}}(M; A, B) := d_{\text{int}}(T_A T_B M, T_B T_A M).$$

Given a window W and tolerance $\eta \geq 0$ declared in `run.yaml`, we say T_A, T_B *soft-commute* on M (on W) if $\Delta_{\text{comm}}(M; A, B) \leq \eta$. Otherwise, a canonical order is fixed and the residual Δ_{comm} is logged into the δ -ledger as an algorithmic defect.

Declaration H.10 (Operational A/B policy (window-coherent)). Per window W and degree i :

1. Compute Δ_{comm} *after* the mandated B-side collapse $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$ and on the same W, τ as the gate and ledger.
2. If $\Delta_{\text{comm}} \leq \eta$, treat T_A, T_B as soft-commuting on that window and allow either order.
3. If $\Delta_{\text{comm}} > \eta$, enforce a canonical order, and log Δ_{comm} as $\delta_{\text{comm}}^{\text{alg}}$ for W, i, τ .
4. Aggregate $\delta_{\text{comm}}^{\text{alg}}$ into the per-window budget $\Sigma\delta$ used by Restart/Summability (Appendix J).

Remark H.11 (Three or more nonnested axes). For A, B, C pairwise soft-commuting does not imply global confluence. Operationally, fix a canonical total order on axes and measure/log adjacent commutation defects along that order; do not infer higher coherence without explicit 2-cell data.

K.4. Mirror/Transfer 2-cells and additive budget (persistence-level accounting)

2-cells as measured defects. Let *Mirror* be an admissible endofunctor on the filtered or persistence layer, equipped with a natural 2-cell

$$\theta : \text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$$

(on the filtered layer, understood up to f.q.i.), whose image under \mathbf{P}_i is measured in d_{int} by a bound $\delta^{\text{mirror}}(i, \tau) \geq 0$ on the current window W .

Proposition H.12 ([Spec] Additive budget with (co)reflectors and Mirror). *On a fixed window W and degree i , the total slack required to accommodate:*

1. *each Mirror–Collapse defect $\delta^{\text{mirror}}(i, \tau)$, and*
2. *each recorded A/B commutation defect $\delta_{\text{comm}}^{\text{alg}} = \Delta_{\text{comm}}(M; A, B)$,*

is bounded above by their quantale-aggregation (default: ordinary sum) in the δ -ledger. Any subsequent 1-Lipschitz persistence-level post-processing does not increase this bound.

Proof sketch. Each defect is a d_{int} -bound for a natural transformation or commutator comparison. Triangle inequalities and 1-Lipschitzness of allowed post-processing yield subadditivity and non-expansive propagation of bounds. Ledger aggregation is required to apply a single law exactly once (Appendix G). \square

Corollary H.13 (Windowwise additivity). *Across a MECE family of windows $\{W_j\}$ with per-window contributions δ_j^{mirror} and δ_j^{comm} , the end-to-end slack is bounded by the ledger aggregation of $\{\delta_j^{\text{mirror}}, \delta_j^{\text{comm}}\}_j$, and this bound is preserved under 1-Lipschitz aggregators and the Restart/Summability mechanism (Appendix J).*

K.5. Strict 2-cell accounting via product quantales (no double counting)

We strictify multi-channel 2-cell budget aggregation by declaring a base quantale and its finite products, and enforcing a single aggregation law applied exactly once (Appendix G, Declaration D.1).

Definition H.14 (Base quantale). A *quantale* (budget object) is a commutative, monotone monoid $(V, \oplus, \preceq, 0)$, where \preceq is a partial order compatible with \oplus . Default instance: $V = [0, \infty]$, $\oplus = +$, $\preceq = \leq$, $0 = 0$. Each pipeline step emits a defect value in V (e.g. δ^{alg} , δ^{disc} , δ^{meas}), and per-window budgets use the *single* operation \oplus .

Definition H.15 (Product quantale and collapse homomorphism). For $m \geq 1$ defect channels, define the *product quantale*

$$V^{\times m} \quad \text{with} \quad \mathbf{x} \preceq \mathbf{y} \iff (\forall r) x_r \preceq y_r, \quad (\mathbf{x} \widehat{\oplus} \mathbf{y})_r := x_r \oplus y_r, \quad \mathbf{0} := (0, \dots, 0).$$

Define the *collapse homomorphism* $\pi : V^{\times m} \rightarrow V$ by

$$\pi(x_1, \dots, x_m) := x_1 \oplus \dots \oplus x_m.$$

Then π is monotone and $\pi(\mathbf{x} \widehat{\oplus} \mathbf{y}) = \pi(\mathbf{x}) \oplus \pi(\mathbf{y})$.

Definition H.16 (AWFS/2-cell bounds (contract)). [Spec/Contract] Assume an algebraic weak factorization system (AWFS) on the filtered layer producing:

1. a comonad L (cofibration-like) with counit $L \Rightarrow \text{Id}$,
2. a monad R (fibration-like) with unit $\text{Id} \Rightarrow R$,
3. 2-cell contracts for admissible functors, including $\mathbf{G}_\tau \Rightarrow \mathbf{G}_\tau^2$, $\text{Mirror} \circ \mathbf{G}_\tau \Rightarrow \mathbf{G}_\tau \circ \text{Mirror}$, and adjacent reflector commutators.

Each contract emits a per-degree/per-threshold defect vector

$$\delta(i, \tau) := (\delta^{\text{mirror}}(i, \tau), \delta^{\text{transfer}}(i, \tau), \delta^{\text{comm}}(i, \tau)) \in V^{\times m},$$

measured after $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$ on the current window W .

Proposition H.17 (Strict product accounting). *Along any pipeline segment with component defect vectors $\delta_1, \dots, \delta_n \in V^{\times m}$, the accumulated segment defect satisfies*

$$\delta_{\text{seg}} \preceq \delta_1 \widehat{\oplus} \dots \widehat{\oplus} \delta_n, \quad \delta_{\text{seg}} := \pi(\delta_{\text{seg}}) \preceq \pi(\delta_1) \oplus \dots \oplus \pi(\delta_n).$$

Thus: (i) each channel is aggregated exactly once (coordinatewise), (ii) the collapsed scalar budget δ_{seg} respects the same single law \oplus , and (iii) double counting is structurally prevented by the homomorphism π .

Proof. Each defect channel is bounded in d_{int} and obeys a triangle inequality, hence the coordinatewise sum $\widehat{\oplus}$ bounds the accumulated vector. Applying π and using $\pi(\mathbf{x} \widehat{\oplus} \mathbf{y}) = \pi(\mathbf{x}) \oplus \pi(\mathbf{y})$ yields the scalar bound. \square

Remark H.18 (run.yaml alignment; determinism). Record in run.yaml (Appendix G, Declaration D.1):

1. quantale.{name,op,unit,order} and layered_delta.compose: "quantale-sum",
2. channel list and the product dimension m ,
3. per-step and per-window vectors $\delta \in V^{\times m}$ and their scalar collapse $\pi(\delta)$,
4. reflector order decisions and soft-commuting verdicts, including η and Δ_{comm} .

This enforces deterministic aggregation and prevents inadvertent re-aggregation.

K.6. Windowed usage and minimal schema (operational contract)

Per window W (degree i , threshold τ), log at minimum:

1. reflector axis list and canonical order; A/B soft-commuting checks with η and measured Δ_{comm} ,
2. admissible 2-cells (Mirror/Transfer) and their measured bounds $\delta^{\text{mirror}}(i, \tau)$, $\delta^{\text{transfer}}(i, \tau)$,
3. the defect vector $\delta(i, \tau) \in V^{\times m}$ and its scalar collapse $\pi(\delta(i, \tau))$,
4. quantale parameters and the aggregation law used exactly once,
5. B-Gate⁺ gap and the resulting per-window budget $\Sigma\delta(i)$ for Restart/Summability (Appendix J).

Canonical YAML keys appear in Appendix G.

K.7. Formalization stubs (Lean/Coq) [Spec]

-- Schematic: reflectors, commutation defect, and quantale products

```
namespace AK.K
open scoped Classical
noncomputable section
```

```
-- Persistence reflectors (schematic)
constant Pers : Type
constant Reflector : Type
constant T : Reflector → (Pers → Pers)
```

```
-- Interleaving distance (schematic)
constant d_int : Pers → Pers → ℝ≥0
```

```

-- Commutation defect
def comm_defect (M : Pers) (A B : Reflector) :  $\mathbb{R}_{\geq 0}$  :=
  d_int (T A (T B M)) (T B (T A M))

-- Quantale
structure Quantale :=
  (V : Type) (op : V → V → V) (le : V → V → Prop) (unit : V)
  (op_comm : ∀ x y, op x y = op y x)
  (op_assoc : ∀ x y z, op (op x y) z = op x (op y z))
  (op_mono : ∀ {x x' y y'}, le x x' → le y y' → le (op x y) (op x' y'))
  (le_refl : ∀ x, le x x) (le_trans : ∀ x y z, le x y → le y z → le x z)

-- Product quantale and collapse homomorphism (fold)
def prodQ (Q : Quantale) (m : Nat) : Quantale := by
  -- coordinatewise order and op
  admit

def collapse (Q : Quantale) (m : Nat) :
  (prodQ Q m).V → Q.V := by
  -- fold by Q.op along coordinates; homomorphism proof required
  admit
end AK.K

```

K.8. Guard-rails and failure modes

- *Nonexact steps.* Heuristic, nonexact truncations may violate monad/comonad laws and stability; exclude them from the reflector class or log them as nonexact with separate handling.
- *Window mismatch.* A/B commutation tests and Mirror–Collapse measurements must use the same window/ τ as the gate and δ -ledger; otherwise the audit is invalid.
- *Multi-axis nonconfluence.* For ≥ 3 nonnested axes, pairwise soft-commuting does not imply global coherence; fix a canonical order and log adjacent defects.
- *Adaptive thresholds.* If τ adapts online, every ledger entry must carry the in-force τ ; do not retroactively reassign defects to different τ .
- *f.q.i. boundaries.* Any statement involving \mathbf{G}_τ or AWFS is [Spec] unless discharged; persistence-layer statements remain authoritative.

Summary. On $\text{Pers}_k^{\text{cons}}$, collapse is governed by the exact, idempotent, 1-Lipschitz monad $\mathbf{M}_\tau = \iota_\tau \mathbf{T}_\tau$ (Theorem H.1, Proposition H.2). On $\text{Ho}(\text{FiltCh}(k))$, an implementable idempotent comonad $\mathbf{G}_\tau = \iota C_\tau^{\text{comb}}$ is recorded as [Spec], required to be compatible with persistence after \mathbf{P}_i (Theorem H.5, Proposition H.6). For multiple reflector axes, nesting yields strict commutation; otherwise an A/B soft-commuting policy controls order via the commutation defect Δ_{comm} and logs residuals into the δ -ledger (Definition H.9, Declaration H.10). Mirror/Transfer 2-cells contribute additional measured defects. All multi-channel defects are aggregated in a product quantale and collapsed by a single homomorphism π , enforcing strict, non-duplicative accounting aligned with `run.yaml` (Proposition H.17, Remark H.18). This integrates with Restart/Summability pasting and overlap auditing (Appendices G and J) under the mandatory after-collapse order.

Appendix L. Quantitative Commutation for Mirror/Tropical [Spec + Pipeline Budget + A/B Policy] (reinforced)

Standing conventions. We work over a field k . All persistence modules are taken in the one-parameter constructible persistence category $\text{Pers}_k^{\text{cons}}$ (locally finite on bounded windows). Filtered (co)limits, when invoked, are computed in $[\mathbb{R}, \text{Vect}_k]$ under the scope policy of Appendix A, Remark .67; returning to the constructible range is permitted only under the logged hypotheses of Appendix A and the window/definability policies of Appendix G. Distances are measured by the interleaving metric d_{int} (which agrees with bottleneck distance on barcodes in the 1D constructible case). The bar-deletion reflector $\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \rightarrow \text{Pers}_k^{\text{cons}}$ is exact and 1-Lipschitz (Appendix A). On the filtered layer we write C_τ for the (filtered) lift of \mathbf{T}_τ , defined only up to filtered quasi-isomorphism (f.q.i.) in $\text{Ho}(\text{FiltCh}(k))$, with endpoints/infinite-bar policy centralized as declared in Chapter 2 and Appendix G. Global conventions: Ext^1 is always against $k[0]$; the energy exponent satisfies $\alpha > 0$ (default $\alpha = 1$); windows are MECE and right-open with overlap lists logged (Appendix G). All measurements and comparisons are performed *after* the mandated order

$$\boxed{\text{for each } t \Rightarrow \mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare}}$$

on the B-side single layer (Appendix G). Any statement not reduced to this persistence-layer order is *[Spec]*.

L.1. Hypotheses (Mirror/Tropical) [Spec]

Spec context. This appendix is *[Spec]*: it records quantitative commutation assumptions and the resulting budgeting rules used by the pipeline. The authoritative, proven layer is the persistence-layer \mathbf{T}_τ and its 1-Lipschitz behavior (Appendix A); all filtered-layer commutation is accepted only through persistence-layer measurements after $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$.

[Spec] Let U be a filtered-level endofunctor (“Mirror/Tropical”) on $\text{Ho}(\text{FiltCh}(k))$. Via \mathbf{P}_i , it induces endofunctors on $\text{Pers}_k^{\text{cons}}$. Assume:

(H1) **1-Lipschitz after persistence.** For each degree i ,

$$d_{\text{int}}(\mathbf{P}_i(UF), \mathbf{P}_i(UG)) \leq d_{\text{int}}(\mathbf{P}_i(F), \mathbf{P}_i(G)) \quad (\forall F, G).$$

(H2) **δ -controlled natural 2-cell (up to f.q.i.).** There exists a natural 2-cell

$$\theta : U \circ C_\tau \Rightarrow C_\tau \circ U$$

in $\text{Ho}(\text{FiltCh}(k))$ (interpreted up to f.q.i.) such that, uniformly in F ,

$$d_{\text{int}}(\mathbf{P}_i(U(C_\tau F)), \mathbf{P}_i(C_\tau(UF))) \leq \delta(i, \tau),$$

for all $i \in \mathbb{Z}$ and $\tau \geq 0$, and where the measurement is taken on the B-side after-collapse layer (Appendix G).

Remark I.1 (Strict commutation). If $\delta(i, \tau) = 0$, then U and C_τ commute up to isomorphism *after* \mathbf{P}_i at the truncated persistence layer, for the declared window/ τ .

Remark I.2 (Tropical instances (Spec)). A “Tropical” operator may be any window-coherent, order-preserving endofunctor assembled from idempotent semiring primitives (e.g. min/max filters, morphological erosions/dilations, monotone reparametrizations) that satisfies (H1) and admits a 2-cell θ with a bound $\delta_{\text{trop}}(i, \tau)$ independent of F . Its configuration (e.g. bin width/range and bin-shift policy) is logged under tropical in `run.yaml` (Appendix G, Declaration D.1).

L.2. Post-collapse bound from the 2-cell (direct persistence form)

Theorem I.3 (Post-collapse non-expansive commutation bound). *Assume (H2). Then for all filtered inputs F , degrees i , and thresholds τ ,*

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(U(C_\tau F)), \mathbf{T}_\tau \mathbf{P}_i(C_\tau(UF))) \leq \delta(i, \tau).$$

Moreover, any further B-side post-processing Φ that is 1-Lipschitz for d_{int} satisfies

$$d_{\text{int}}(\Phi \mathbf{T}_\tau \mathbf{P}_i(U(C_\tau F)), \Phi \mathbf{T}_\tau \mathbf{P}_i(C_\tau(UF))) \leq \delta(i, \tau).$$

Proof. Apply \mathbf{T}_τ , which is 1-Lipschitz, to the (H2) estimate at the persistence layer, and then use 1-Lipschitzness of Φ . \square

Corollary I.4 (Two-stage additivity). *If U_1, U_2 satisfy (H2) with bounds δ_1, δ_2 , then for all F, i, τ ,*

$$d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(U_2 U_1(C_\tau F)), \mathbf{T}_\tau \mathbf{P}_i(C_\tau(U_2 U_1 F))) \leq \delta_1(i, \tau) + \delta_2(i, \tau).$$

Remark I.5 (Necessity of a controlled 2-cell). Without a controlled 2-cell, 1-Lipschitzness of U and \mathbf{T}_τ does not constrain the discrepancy between $U \circ C_\tau$ and $C_\tau \circ U$. In particular, near- τ accumulations can create discontinuous behavior invisible to naive nonexpansivity checks, and must be treated as Type IV risk unless explicitly bounded (Appendix D).

L.3. Product-ledger budgeting (direct accumulation; no relay chains)

We mainstream direct accumulation of commutation errors in a product-quantale ledger (Appendix K, §K.5), eliminating intermediate relay comparisons and preventing double counting.

Definition I.6 (Base and product quantales). Fix a commutative quantale (budget object) $(V, \oplus, \preceq, 0)$ (default: $V = [0, \infty]$, $\oplus = +$, $\preceq = \leq$). For $m \geq 1$ channels (e.g. Mirror, Transfer, A/B), set the product quantale

$$V^{\times m}, \quad \mathbf{x} \preceq \mathbf{y} \iff (\forall r) x_r \preceq y_r, \quad (\mathbf{x} \widehat{\oplus} \mathbf{y})_r := x_r \oplus y_r, \quad \mathbf{0} := (0, \dots, 0).$$

Let $\pi : V^{\times m} \rightarrow V$ be the collapse homomorphism $\pi(\mathbf{x}) = x_1 \oplus \dots \oplus x_m$ (Appendix K).

Definition I.7 (Per-window defect vector). Fix a window $W = [u, u')$ (MECE, right-open), a degree i , and the threshold(s) τ in force on W . Each Mirror/Tropical step U_j on W contributes a vector $\delta_W^{(j)}(i) \in V^{\times m}$ whose nonzero coordinates are the relevant bounds furnished by Theorem I.3 (e.g. mirror/transfer coordinates). Each A/B commutation failure contributes a vector whose only nonzero coordinate is the A/B channel value equal to the measured Δ_{comm} (Definition I.11). The window vector and its scalar collapse are

$$\delta_W(i) := \widehat{\oplus}_j \delta_W^{(j)}(i), \quad \Sigma \delta_W(i) := \pi(\delta_W(i)).$$

All entries must be logged with the window id, degree, and the in-force τ (Appendix G).

Proposition I.8 (Strict, non-duplicative accounting). *Coordinatewise aggregation in $V^{\times m}$ followed by π yields*

$$\Sigma \delta_W(i) \preceq \bigoplus_j \pi(\delta_W^{(j)}(i)),$$

counts each channel exactly once, and is compatible with Appendix K, Proposition H.17.

Proof. Immediate from the definition of $\widehat{\oplus}$ (coordinatewise application of \oplus) and the homomorphism property of π . \square

L.4. Windowed pipeline bound (direct form)

Theorem I.9 ([Spec] Direct pipeline bound). *Fix a window W , degree i , and the realized left/right orderings $\Pi_{\text{lhs}}, \Pi_{\text{rhs}}$ on W . Assume that every commutation step in the segment is covered either by: (i) a recorded 2-cell bound (Mirror/Tropical/Transfer) contributing a coordinate to $\delta_W(i)$, or (ii) an A/B commutation test contributing a coordinate to $\delta_W(i)$. Then for any filtered input F ,*

$$d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(\Pi_{\text{lhs}}(F)), \mathbf{T}_\tau \mathbf{P}_i(\Pi_{\text{rhs}}(F))\right) \preceq \Sigma \delta_W(i),$$

where τ is the threshold in force on W . The bound is uniform in F , additive across steps via $\widehat{\oplus}$, and non-increasing under any subsequent 1-Lipschitz persistence post-processing.

Proof sketch. Each step discrepancy is bounded in the appropriate channel either by Theorem I.3 (2-cell) or by the A/B commutation measurement. Triangle inequalities aggregate these bounds coordinatewise, giving $\delta_W(i)$. Applying π yields the scalar bound $\Sigma \delta_W(i)$. Finally, 1-Lipschitz post-processing cannot increase the discrepancy. \square

Corollary I.10 (Across windows; Restart/Summability alignment). *Over a MECE partition $\{W_k\}$,*

$$\bigoplus_k \Sigma \delta_{W_k}(i)$$

controls the end-to-end discrepancy for degree i . Compare this per-degree accumulated budget to the per-window safety margins $\text{gap}_{\tau_k}(i)$ to apply Restart and Summability (Appendix J).

L.5. Operational A/B policy for reflector axes (soft-commuting)

Definition I.11 (A/B commutation test). Let T_A, T_B be exact reflectors on $\text{Pers}_k^{\text{cons}}$ (Appendix K). For $M \in \text{Pers}_k^{\text{cons}}$, define

$$\Delta_{\text{comm}}(M; A, B) := d_{\text{int}}(T_A T_B M, T_B T_A M).$$

Given a tolerance $\eta \geq 0$ declared per window W in `run.yaml`, declare *soft-commuting* if $\Delta_{\text{comm}} \leq \eta$. If $\Delta_{\text{comm}} > \eta$, fix a canonical order and ledger the residual as the A/B coordinate in $\delta_W(i)$. Skip testing for nested torsions where $T_A \circ T_B = T_{A \vee B}$ holds strictly (Appendix K, Proposition H.8). All tests are performed after $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$ on the window W .

L.6. Tropical specification hooks and run.yaml alignment

For reproducibility and audit (Appendix G, Declaration D.1), record at minimum:

- tropical: configuration (e.g. `bins.{width,range}`) and `policy:"after__collapse__only"`.
- `awfs__2cell.two__cell__bounds`: per-channel bounds (mirror/transfer) used as coordinates of δ .
- Per window/degree, record both the vector and its scalar collapse $\Sigma \delta_W(i) = \pi(\delta_W(i))$, together with the window id and in-force τ .

Remark I.12 (Illustrative YAML snippet). tropical:

```
bins: { width: 0.05, range: [0.0, 2.0] }
policy: "after__collapse__only"
awfs__2cell:
  two__cell__bounds:
    mirror__collapse: 0.006
```

```

    transfer_collapse: 0.004
window:
  id: "W03"
  degree: 1
  tau: 0.40
  budget:
    delta_vector: { mirror: 0.006, transfer: 0.004, AB: 0.002 }
    sum_delta: 0.012 #  $\pi(\text{delta\_vector})$  under the declared quantale
  b_gate_plus:
    passed: true
    gap_tau: 0.09

```

L.7. Minimal pseudocode (product-ledger, direct)

```

# steps on window W, degree i; each step carries a coordinate vector in  $V^{\times m}$ 
def window_budget_vector(steps, i):
    vec = {"mirror": 0.0, "transfer": 0.0, "AB": 0.0} #  $V=[0,\infty]$ ,  $\oplus=+$ 
    for s in steps:
        kind = s["kind"] # "mirror" | "transfer" | "AB"
        delta = s["delta"] # measured after  $P_i \rightarrow T_\tau$  on window W
        vec[kind] += delta # coordinatewise  $\oplus$ 
    return vec

def collapse_quantale(vec):
    #  $\pi$ : product  $\rightarrow$  base quantale (here: sum of coordinates)
    return vec["mirror"] + vec["transfer"] + vec["AB"]

```

L.8. Edge cases and guard-rails

- *No (H2), no bound.* Without a controlled 2-cell, no quantitative commutation estimate exists; treat the step as unbudgeted and reject (or mark as Type IV risk) unless separately audited.
- *Window/ τ coherence.* All measurements must use the same window and in-force τ as B-Gate⁺ and the δ -ledger; otherwise the audit is invalid.
- *Per-degree reporting.* Log per-degree vectors $\delta_W(i)$ and per-degree totals $\Sigma \delta_W(i)$; do not rely only on cross-degree sums, as small residuals may hide in totals.
- *Multiple nonnested axes.* Pairwise soft-commuting does not imply global confluence; fix a canonical order, test adjacent pairs, and ledger residuals (Appendix K, Remark H.11).
- *Adaptive thresholds.* If τ adapts online, every defect entry must carry the τ in force at measurement time; do not retroactively migrate defects across τ .

Summary. Assuming a Mirror/Tropical operator U admits a $\delta(i, \tau)$ -controlled 2-cell against the collapse lift C_τ , the post-collapse discrepancy between $U \circ C_\tau$ and $C_\tau \circ U$ is bounded by $\delta(i, \tau)$ and remains bounded under further 1-Lipschitz B-side post-processing (Theorem I.3). Commutation costs are accumulated directly per window in a product-quantale δ -ledger and collapsed exactly once by a homomorphism π , preventing double counting and enabling deterministic auditing (Definitions I.6–I.7, Proposition I.8). The resulting scalar $\Sigma \delta_W(i)$ integrates with B-Gate⁺, Restart, and Summability (Appendix J), while Tropical configuration, 2-cell bounds, window ids, degrees, and in-force τ are recorded in `run.yaml` per Appendix G.

Appendix M. (Optional) Lax Monoidal Compatibility [Spec + Windowed Usage + Budget Integration] (reinforced)

Status (canon). Statements tagged *[Spec]* are operational, windowed contracts. All *strict equalities and invariants* are asserted at the persistence layer in the one-parameter constructible category $\text{Pers}_k^{\text{cons}}$. Chain-/filtered-layer claims are used only *up to filtered quasi-isomorphism (f.q.i.)* in $\text{Ho}(\text{FiltCh}(k))$ and are accepted only insofar as they reduce to persistence-layer measurements after the mandatory order

$$\boxed{\text{for each } t \Rightarrow \mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare}}$$

on the B-side single layer (Appendix G).

M.0. Standing conventions and scope

Ground field and categories. Fix a field k . Let $\text{Pers}_k^{\text{cons}}$ denote the category of constructible (1-parameter) persistence modules $M : \mathbb{R} \rightarrow \text{Vect}_k$, i.e. pointwise finite-dimensional with locally finite critical parameters on bounded windows (equivalently, locally finite barcodes). Filtered chain complexes are denoted $F = (F^t)_{t \in \mathbb{R}}$ with each F^t a bounded (on windows) chain complex of finite-dimensional k -vector spaces and locally finite changes in t .

Scope policy for (co)limits. Whenever filtered (co)limits are invoked, they are computed in the functor category $[\mathbb{R}, \text{Vect}_k]$ under the scope policy of Appendix A, Remark .67, and only then (when explicitly stated) returned to the constructible range under the logged window/definability hypotheses (Appendix G).

Interleaving metric and truncation. We use the interleaving metric d_{int} on $\text{Pers}_k^{\text{cons}}$ (bottleneck distance on barcodes in the constructible 1D case). For each $\tau > 0$, $\mathbf{T}_\tau : \text{Pers}_k^{\text{cons}} \rightarrow \text{Pers}_k^{\text{cons}}$ denotes the exact bar-deletion reflector (Appendix A): it is exact and 1-Lipschitz for d_{int} . On the filtered layer, C_τ denotes a chosen lift of \mathbf{T}_τ , defined only up to f.q.i. in $\text{Ho}(\text{FiltCh}(k))$; endpoint/infinite-bar policies are centralized (Chapter 2, Appendix G). No claim of strict functorial equality $\mathbf{P}_i(C_\tau F) = \mathbf{T}_\tau(\mathbf{P}_i F)$ is made unless reduced to persistence-layer comparison and logged as a policy claim.

Windowing and measurability. Fix MECE, right-open windows on $[0, \infty)$: $\mathcal{W} = \{W_j = [w_j, w_{j+1})\}_{j \in J}$, locally finite on bounded ranges (Appendix G). For any $M \in \text{Pers}_k^{\text{cons}}$, the rank function $t \mapsto \dim_k M(t)$ is piecewise constant with locally finite jumps on bounded windows, hence measurable and integrable.

Energy functional (windowed). For $\sigma > 0$ and a persistence module M , define the clipped rank integral

$$\text{PE}^{\leq \sigma}(M) := \int_0^\sigma \dim_k M(t) dt. \quad (\text{J.1})$$

For a filtered complex F and degree i , define

$$\text{PE}_i^{\leq \sigma}(F) := \text{PE}^{\leq \sigma}(\mathbf{P}_i(F)) = \int_0^\sigma \beta_i(F; t) dt, \quad \beta_i(F; t) := \dim_k \mathbf{P}_i(F)(t). \quad (\text{J.2})$$

All integrals are Lebesgue integrals; constructibility implies they reduce to finite sums on bounded windows.

Pointwise tensor. On persistence modules and filtered complexes we use the pointwise tensor:

$$(M \otimes N)(t) := M(t) \otimes_k N(t), \quad (F \otimes G)^t := F^t \otimes_k G^t.$$

All tensors are over k .

[Spec] scope. This appendix provides windowed, reproducible contracts about how collapse interacts with tensor *laxly*. No assertion of strict symmetric monoidality for C_τ (or \mathbf{T}_τ) is made beyond what is explicitly stated and (when needed) measured after $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$.

M.1. Hypotheses and the laxator

We work under the following hypotheses.

(M1) Exactness and constructibility of pointwise tensor. The pointwise tensor \otimes on Vect_k is exact and biadditive; hence the induced pointwise tensor on $[\mathbb{R}, \text{Vect}_k]$ is exact in each variable. Moreover, for $M, N \in \text{Pers}_k^{\text{cons}}$, the tensor $M \otimes N$ lies in $\text{Pers}_k^{\text{cons}}$ on bounded windows.

(M2) Künneth over a field (pointwise, on windows). For filtered complexes F, G satisfying the window-bounded finiteness assumptions above, for each $t \in \mathbb{R}$ and each $i \in \mathbb{Z}$,

$$H_i((F \otimes G)^t) \cong \bigoplus_{p+q=i} H_p(F^t) \otimes_k H_q(G^t),$$

naturally in (F, G) and t . Equivalently (after assembling in t), in $\text{Pers}_k^{\text{cons}}$ one has a natural isomorphism

$$\mathbf{P}_i(F \otimes G) \cong \bigoplus_{p+q=i} \mathbf{P}_p(F) \otimes \mathbf{P}_q(G), \quad (\text{J.3})$$

on any bounded window where both sides are constructible.

(M3) Lax compatibility for collapse (filtered layer) [Spec]. *[Spec]* There exists a natural transformation (the laxator), in $\text{Ho}(\text{FiltCh}(k))$ up to f.q.i.,

$$\lambda_{\tau, F, G} : C_\tau(F \otimes G) \Longrightarrow C_\tau F \otimes C_\tau G, \quad (\text{J.4})$$

natural in F, G and τ , whose operational meaning is only through persistence-layer measurements after $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$.

We will occasionally strengthen (M3) to:

(M3⁺) *[Spec]* For all $t \in \mathbb{R}$ and degrees i , the induced map in homology

$$H_i(\lambda_{\tau, F, G}^t) : H_i((C_\tau(F \otimes G))^t) \longrightarrow H_i((C_\tau F \otimes C_\tau G)^t)$$

is a monomorphism (equivalently, ranks do not decrease under λ pointwise in (t, i)).

Remark J.1 (Intervals under tensor). For interval modules over k ,

$$k_{[a, b)} \otimes k_{[c, d)} \cong \begin{cases} k_{[\max\{a, c\}, \min\{b, d\})}, & \text{if } [a, b) \cap [c, d) \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases}$$

Thus tensor intersects lifespans; at the barcode level this is the field Künneth rule.

Remark J.2 (Constructibility under tensor). On any bounded window, only finitely many bars meet; hence the critical set of $M \otimes N$ is locally finite and pointwise dimensions remain finite. Therefore $M \otimes N \in \text{Pers}_k^{\text{cons}}$ on bounded windows.

Remark J.3 (Monotonicity scope). Tensor is neither purely deletion-type nor purely inclusion-type in general. Accordingly, we confine monotonicity claims to: (i) windowed energy bounds, and (ii) regimes where a monomorphic laxator ($M3^+$) is verified or where a persistence-layer injection is used (Section M.3⁺).

M.2. Windowed energy via overlap integrals

Definition J.4 (Windowed seminorms). For $M \in \text{Pers}_k^{\text{cons}}$ and window $[0, \sigma]$, define

$$\|M\|_{E_1}^{(\sigma)} := \text{PE}^{\leq \sigma}(M), \quad \|M\|_{\infty}^{(\sigma)} := \sup_{t \in [0, \sigma]} \dim_k M(t).$$

For a filtered complex F and degree i , set $\|F\|_{E_1}^{(i, \sigma)} := \|\mathbf{P}_i(F)\|_{E_1}^{(\sigma)}$ and $\|F\|_{\infty}^{(i, \sigma)} := \|\mathbf{P}_i(F)\|_{\infty}^{(\sigma)}$.

Theorem J.5 (Convolution identity (K nneth) and upper bounds). Assume (M1)–(M2). Then for all filtered complexes F, G , degrees i , and $\sigma > 0$,

$$\text{PE}_i^{\leq \sigma}(F \otimes G) = \sum_{p+q=i} \int_0^{\sigma} \beta_p(F; t) \beta_q(G; t) dt, \quad (\text{J.5})$$

where the sum is finite on windows due to bounded homological range. Consequently,

$$\text{PE}_i^{\leq \sigma}(F \otimes G) \leq \sum_{p+q=i} \left(\sup_{t \in [0, \sigma]} \beta_q(G; t) \right) \text{PE}_p^{\leq \sigma}(F), \quad (\text{J.6})$$

and symmetrically with $(F, p) \leftrightarrow (G, q)$. Equivalently, in seminorm form,

$$\|F \otimes G\|_{E_1}^{(i, \sigma)} \leq \sum_{p+q=i} \|F\|_{E_1}^{(p, \sigma)} \|G\|_{\infty}^{(q, \sigma)}, \quad \|F \otimes G\|_{E_1}^{(i, \sigma)} \leq \sum_{p+q=i} \|F\|_{\infty}^{(p, \sigma)} \|G\|_{E_1}^{(q, \sigma)}.$$

Proof. By (J.3), $\dim_k \mathbf{P}_i(F \otimes G)(t) = \sum_{p+q=i} \dim_k \mathbf{P}_p(F)(t) \cdot \dim_k \mathbf{P}_q(G)(t)$ for each t . Integrate over $[0, \sigma]$ to obtain (J.5). Bounding one factor by its sup on $[0, \sigma]$ gives (J.6) and the seminorm inequalities. \square

Remark J.6 (Barcode interpretation). In the interval-decomposable case (1D constructible over a field), $\beta_i(F; t)$ counts i -bars alive at t . The integrand $\beta_p(F; t) \beta_q(G; t)$ counts ordered pairs of alive bars; integrating sums lengths of pairwise intersections, matching “tensor = intersection” (Remark J.1).

M.2*. Clip/contract stability (deletion-type monotonicity)

Proposition J.7 (Clipping and collapse are nonincreasing). For any $\sigma > 0$:

1. Clipping. Restricting to a subwindow (MECE, right-open) does not increase $\| - \|_{E_1}^{(\sigma)}$ nor $\| - \|_{\infty}^{(\sigma)}$, and leaves values unchanged on $[0, \sigma]$.
2. Bar-deletion collapse. For any $\tau > 0$ and $M \in \text{Pers}_k^{\text{cons}}$,

$$\|\mathbf{T}_{\tau} M\|_{E_1}^{(\sigma)} \leq \|M\|_{E_1}^{(\sigma)}, \quad \|\mathbf{T}_{\tau} M\|_{\infty}^{(\sigma)} \leq \|M\|_{\infty}^{(\sigma)}.$$

Consequently, for filtered F and degree i , $\|\mathbf{T}_{\tau} \mathbf{P}_i(F)\|_{E_1}^{(\sigma)} \leq \|\mathbf{P}_i(F)\|_{E_1}^{(\sigma)}$ and similarly for $\| \cdot \|_{\infty}$.

Proof. Clipping is restriction. Bar deletion implies pointwise $\dim(\mathbf{T}_{\tau} M)(t) \leq \dim M(t)$ on any window; integrate and take sup. \square

M.3. Collapse vs tensor: laxity and energy dominance

Proposition J.8 (Collapsed convolution bound). *Assume (M1)–(M2). For any $\tau, \sigma > 0$ and degree i ,*

$$\text{PE}^{\leq \sigma}(\mathbf{T}_\tau \mathbf{P}_i(F \otimes G)) \leq \text{PE}^{\leq \sigma}(\mathbf{P}_i(F \otimes G)) = \sum_{p+q=i} \int_0^\sigma \beta_p(F; t) \beta_q(G; t) dt.$$

Proof. Apply Proposition J.7 to $\mathbf{P}_i(F \otimes G)$, then Theorem J.5. \square

Theorem J.9 (Energy dominance under a monomorphic laxator [Spec]). *Assume (M3) and (M3⁺) [Spec]. Then for all $\tau, \sigma > 0$ and degrees i ,*

$$\text{PE}_i^{\leq \sigma}(C_\tau(F \otimes G)) \leq \text{PE}_i^{\leq \sigma}(C_\tau F \otimes C_\tau G),$$

and therefore, by Theorem J.5 applied on the right-hand side where defined,

$$\text{PE}_i^{\leq \sigma}(C_\tau(F \otimes G)) \leq \sum_{p+q=i} \int_0^\sigma \beta_p(C_\tau F; t) \beta_q(C_\tau G; t) dt.$$

Proof. By (M3⁺), $\beta_i(C_\tau(F \otimes G); t) \leq \beta_i(C_\tau F \otimes C_\tau G; t)$ for all $t \in [0, \sigma]$. Integrate. \square

M.3[†]. Persistence-layer ECF low-pass laxator (explicit) [Spec]

Definition (ECF low-pass injection). [Spec] In the interval regime (1D constructible), view \mathbf{T}_τ as deleting all bars of lifespan $\leq \tau$ and keeping all longer bars unchanged (no shrinking). Define a natural transformation

$$\lambda_{\tau, M, N}^{\text{ECF}} : \mathbf{T}_\tau(M \otimes N) \longrightarrow (\mathbf{T}_\tau M) \otimes (\mathbf{T}_\tau N) \quad (\text{J.7})$$

by specifying it on interval summands using $k_I \otimes k_J \cong k_{I \cap J}$ (Remark J.1):

- If $\ell(I \cap J) \leq \tau$, then the source summand is killed by \mathbf{T}_τ , hence maps to 0.
- If $\ell(I \cap J) > \tau$, then necessarily $\ell(I) > \tau$ and $\ell(J) > \tau$; thus $\mathbf{T}_\tau k_I = k_I$, $\mathbf{T}_\tau k_J = k_J$, and the map on the $k_{I \cap J}$ summand is the identity inclusion into $k_I \otimes k_J \cong k_{I \cap J}$.

Extend additively to finite sums on bounded windows.

Consequences. [Spec] The map $\lambda_{\tau, M, N}^{\text{ECF}}$ is pointwise monomorphic (hence rank-nonincreasing on sources), yielding for any window $[0, \sigma]$:

$$\text{PE}^{\leq \sigma}(\mathbf{T}_\tau(M \otimes N)) \leq \text{PE}^{\leq \sigma}((\mathbf{T}_\tau M) \otimes (\mathbf{T}_\tau N)).$$

When $M = \mathbf{P}_p(F)$, $N = \mathbf{P}_q(G)$, this provides a concrete, window-coherent candidate for the lax behavior used by filtered-level contracts.

Remark J.10 (Use with the canon order). Any use of (J.7) in the pipeline must be evaluated after the canon order $\mathbf{P}_i \rightarrow \mathbf{T}_\tau \rightarrow$ compare on the B-side, with the same window/ τ as the ledger (Appendix G).

M.4. Interaction with τ -sweeps and stability bands

Definition J.11 (τ -sweep for energy). Fix a bounded window $[0, \sigma]$ and an object $M \in \text{Pers}_k^{\text{cons}}$. A τ -sweep is a finite or countable increasing array $\{\tau_\ell\} \subset (0, \infty)$. A subarray is an *energy stability band* if $\dim(\mathbf{T}_\tau M)(t)$ is constant for all $t \in [0, \sigma]$ and all τ in the band (equivalently, no bar length of M lies in the band boundary in a way that affects $[0, \sigma]$).

Proposition J.12 (Piecewise constancy in τ on bounded windows). Fix $M \in \text{Pers}_k^{\text{cons}}$ and $[0, \sigma]$. There exists a finite critical set $S \subset (0, \infty)$ (depending on $M, [0, \sigma]$) such that $\tau \mapsto \text{PE}^{\leq \sigma}(\mathbf{T}_\tau M)$ is constant on each connected component of $(0, \infty) \setminus S$. In particular, energy stability bands are unions of open intervals intersected with the sweep.

Proof. On $[0, \sigma]$, only finitely many bars of M meet the window. \mathbf{T}_τ changes only when τ crosses one of these bar lengths; between such values the deleted/retained set is constant, hence the rank function and its integral are constant. \square

M.5. Budget integration and quantitative gaps [Spec]

Definition J.13 (Monoidal-laxity budget entry). [Spec] On a window $W = [u, u']$, degree i , and threshold τ , define a measurable persistence-layer laxity gap by

$$\delta_W^{\text{lax}}(i, \tau) := d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(C_\tau(F \otimes G)), \mathbf{T}_\tau \mathbf{P}_i(C_\tau F \otimes C_\tau G)\right), \quad (\text{J.8})$$

computed after the canon order and using the same window/ τ as the ledger. If only an upper bound is available (e.g. via ECF injection at the persistence layer), record that upper bound.

Proposition J.14 (Ledger compatibility (quantale/product-quantale)). [Spec] The entry $\delta_W^{\text{lax}}(i, \tau)$ may be recorded as: (i) a scalar defect in the base quantale (V, \oplus) , or (ii) a dedicated coordinate (e.g. `lax_tensor`) in the product quantale $V^{\times m}$ (Appendix K–L), and then aggregated once via the declared quantale law (Appendix G). Any subsequent 1-Lipschitz post-processing on the persistence layer does not increase the recorded bound.

Remark J.15 (run.yaml alignment). Log in run.yaml (Appendix G, Declaration D.1) the window id, degree i , in-force τ , and either: `budget.delta_vector.lax_tensor` (product-ledger) or `budget.sum_delta` (scalar), together with `policy.after_collapse_only=true`. This prevents double counting.

M.6. Edge cases and pitfalls

- *Nonexact tensor surrogates.* If an operation used in place of \otimes is not exact/biadditive, Künneth and the convolution identity may fail; treat all such cases as [Spec] and require measured bounds.
- *Unverified Künneth context.* If the hypotheses for (J.3) fail (e.g. out of the window-bounded regime), do not assert equality (J.5); use only measured or ledgered bounds.
- *Missing monomorphy.* Without (M3⁺) (or without the ECF persistence injection), energy dominance of $C_\tau(F \otimes G)$ by $C_\tau F \otimes C_\tau G$ is not guaranteed.
- *Window/ τ mismatch.* All measurements must use the same MECE right-open window and the in-force τ as B-Gate⁺ and the δ -ledger; otherwise Restart/Summability pasting can fail (Appendix J).

M.7. Worked examples and tests

Example J.16 (Single-interval bars (persistence level)). Let $M = k_{[a,b]}$, $N = k_{[c,d]}$. Then $M \otimes N \cong k_{[a,b] \cap [c,d]}$ (Remark J.1). Hence for any $\sigma > 0$,

$$\text{PE}^{\leq \sigma}(M \otimes N) = \lambda\left(\left([a,b] \cap [c,d]\right) \cap [0,\sigma]\right),$$

where $\lambda(\cdot)$ is Lebesgue measure (length).

Example J.17 (Effect of bar-deletion collapse). If $\ell([a,b]) \leq \tau$ or $\ell([c,d]) \leq \tau$, then $\mathbf{T}_\tau M = 0$ or $\mathbf{T}_\tau N = 0$. Moreover, if $\ell([a,b] \cap [c,d]) \leq \tau$ then $\mathbf{T}_\tau(M \otimes N) = 0$. The ECF injection (J.7) implies $\text{PE}^{\leq \sigma}(\mathbf{T}_\tau(M \otimes N)) \leq \text{PE}^{\leq \sigma}((\mathbf{T}_\tau M) \otimes (\mathbf{T}_\tau N))$.

Proposition J.18 (Test model for (M3⁺) on interval sums). [Spec] Let $F \simeq \bigoplus_r k_{I_r}[-p_r]$ and $G \simeq \bigoplus_s k_{J_s}[-q_s]$ be finite direct sums of interval modules (shifted in homological degree) on a bounded window, and let C_τ implement bar deletion (low-pass) at scale τ . Define $\lambda_{\tau,F,G}$ on summands by the canonical inclusions induced by $\mathbf{T}_\tau(k_{I_r} \otimes k_{J_s}) \hookrightarrow \mathbf{T}_\tau k_{I_r} \otimes \mathbf{T}_\tau k_{J_s}$, extended additively and lifted to complexes up to f.q.i. Then (M3⁺) holds on the window, and energy dominance of Theorem J.9 follows.

M.8. Formal underpinnings (constructibility, measurability, exactness)

Lemma J.19 (Constructibility preserved by tensor). Assume (M1). If $M, N \in \text{Pers}_k^{\text{cons}}$, then $M \otimes N \in \text{Pers}_k^{\text{cons}}$ on bounded windows. If F, G are constructible filtered complexes on windows, then $F \otimes G$ is constructible on windows.

Proof. For persistence modules, use Remark J.2. For filtered complexes, apply the same argument degree-wise on windows and use window-bounded finiteness. \square

Lemma J.20 (Measurability and finiteness of energy). For constructible M , $t \mapsto \dim_k M(t)$ is piecewise constant with locally finite jumps on bounded windows, hence $\text{PE}^{\leq \sigma}(M) < \infty$ for all $\sigma > 0$.

Proof. On a bounded window, constructibility implies finitely many rank changes; the rank function is a finite sum of characteristic functions of intervals. Integrability follows. \square

Lemma J.21 (\mathbf{T}_τ is exact and 1-Lipschitz). \mathbf{T}_τ is exact on short exact sequences in $\text{Pers}_k^{\text{cons}}$ and satisfies $d_{\text{int}}(\mathbf{T}_\tau M, \mathbf{T}_\tau N) \leq d_{\text{int}}(M, N)$ for all M, N .

Proof. This is Appendix A: \mathbf{T}_τ is an exact bar-deletion reflector and 1-Lipschitz for d_{int} . \square

M.9. Operational checklist (windowed, reproducible) [Spec]

For each experiment window $W = [u, u']$ (or $[0, \sigma]$ by translation):

- Log the tensor context: pairs (F, G) , monitored degrees i , and the window id (MECE, right-open).
- Log the in-force τ and confirm the canon order (after $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$).
- State whether (M2) is invoked on the window; otherwise replace Theorem J.5 by measured estimates only.
- If using filtered-layer laxity (M3), specify $\lambda_{\tau,F,G}$ and whether (M3⁺) is verified (e.g. rank checks on sampled t points).

- If a persistence-layer bound is used (Section M.3[†]), log it as `lax_tensor` (product ledger) or as a scalar defect.
- Aggregate defects using the declared quantale law exactly once (Appendix G, Appendix K–L) and compare against $B\text{-Gate}^+$ gaps for Restart/Summability (Appendix J).

M.10. Summary of contracts [*Spec*]

On bounded windows, tensor admits a deterministic energy calculus: under pointwise field Künneth, the windowed energy of $F \otimes G$ in degree i equals an overlap integral of Betti curves (Theorem J.5). Bar-deletion collapse \mathbf{T}_τ is deletion-type and nonincreasing for both E_1 (energy) and L^∞ seminorms (Proposition J.7). Lax monoidal compatibility of collapse is used only as a windowed contract: either via a verified monomorphic laxator ($M3^+$) giving energy dominance (Theorem J.9), or via an explicit persistence-layer injection in the interval regime (Section M.3[†]). Any residual “laxity gap” may be recorded as a quantale-budget defect and integrated with the product-ledger accounting of Appendices K–L and the Restart/Summability pasting of Appendix J.

M.11. Formalization blueprint (Lean/Coq) [*Spec*]

A minimal API for formalization includes:

- `PersCons` (constructible persistence), pointwise tensor tensor with exactness and constructibility lemmas (Lemma J.19).
- A windowed energy functional `PE` on persistence modules and `PEi` on filtered objects via \mathbf{P}_i .
- A Künneth interface yielding (J.3) on windows and the energy identity (J.5).
- The reflector \mathbf{T}_τ with exactness and 1-Lipschitz proofs (Appendix A; Lemma J.21).
- Optional laxator (filtered layer, up to f.q.i.) and predicate `laxator_mono`; optional persistence-layer `ecf_lax` implementing (J.7).
- Budget hooks aligned with `run.yaml` keys: quantale parameters, product-ledger coordinate `lax_tensor`, and single aggregation law.

Appendix N. Projection Formula and Base Change [*Spec* + Windowed Protocol + Budget Integration] (reinforced)

Standing conventions (canon). We work over a coefficient *field* Λ (e.g. a base field k , or at [*Spec*] level a Novikov field), and all statements below are phrased uniformly for Λ . All persistence modules are constructible; all equality/identity claims are asserted at the persistence layer in the one-parameter constructible category $\text{Pers}_\Lambda^{\text{cons}}$. Filtered (co)limits, when used, are computed objectwise in $[\mathbb{R}, \text{Vect}_\Lambda]$ under the scope policy of Appendix A, Remark .67, and only then (when stated) returned to the constructible range; any such return must be logged (Appendix G) and may be enforced by \mathbf{T}_τ . Distances are measured by the interleaving metric d_{int} (= bottleneck in the constructible 1D setting). Truncation \mathbf{T}_τ is the exact bar-deletion reflector and is 1-Lipschitz (Appendix A, Proposition .70); on the filtered-complex side we use C_τ *only up to f.q.i.* in $\text{Ho}(\text{FiltCh}(\Lambda))$, and write \mathbf{P}_i for degree- i persistence. Global conventions: Ext^1 is always taken against $\Lambda[0]$; the energy exponent satisfies $\alpha > 0$ (default $\alpha = 1$); windows are MECE and right-open (Appendix G).

References to “infinite bars/generic dimension” point to Appendix D, Remark A.2. Monotonicity claims follow the global policy: deletion-type only (nonincreasing), inclusion-type merely stable/non-expansive. All comparisons follow the mandatory order

$$\boxed{\text{for each } t \Rightarrow \mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare}}$$

on the B-side single layer, after collapse (Appendix G).

N.1. Hypotheses (PF/BC layer) and normalizations

[Spec] We fix a class of filtered spaces and maps $f : X \rightarrow Y$ for which a six-functor formalism is available on $D_c^b(\text{Shv}_\Lambda(-))$ (bounded derived category of Λ -constructible sheaves), and adopt:

- (N0) **Coefficients and Tor control.** Λ is a field. All objects invoked in tensor expressions have finite Tor-dimension, hence no Tor-corrections appear in Künneth/projection-formula statements.
- (N1) **Constructibility/finiteness.** All sheaves are constructible; we use the standard t -structure on D_c^b ; (co)homology objects are finite dimensional on bounded windows.
- (N2) **PF/BC hypotheses.** Projection formula (PF) and base change (BC) are invoked only under the usual hypotheses:
 - PF: for f proper, $Rf_*(A \otimes^{\mathbf{L}} f^* B) \simeq Rf_* A \otimes^{\mathbf{L}} B$.
 - BC: for a Cartesian square and f proper (or smooth with the appropriate $f^!$ variant), $Lg^* Rf_* A \simeq Rf'_* Lg'^* A$.

- (N3) **Degree normalization and objectwise evaluation in t .** We use *cohomological* indexing on D_c^b and evaluate realizations objectwise in t :

$$\mathcal{R}(F)^t \cong \mathcal{R}(F^t), \quad \mathbf{P}_i(F)(t) \cong H_i(F^t) \cong H^{-i}(\mathcal{R}(F^t)).$$

Hence \mathbf{P}_i reads off the $(-i)$ -th cohomology sheaf along the filtration. Any geometric shift from $f^!$ (smooth case) is absorbed by this bookkeeping and must be logged if used.

- (N4) **Tensor and window legality.** Tensor is pointwise in t : $(A \otimes^{\mathbf{L}} B)^t \cong A^t \otimes^{\mathbf{L}} B^t$, exact over the field Λ . Constructibility on bounded windows is preserved; integrals and event counts on bounded windows are justified by Appendix H (Tonelli/finite-event regime).

Remark K.1 (Scope and return to constructible). All PF/BC comparisons below are formed in the derived category, evaluated objectwise in t , and then passed to the persistence layer via \mathbf{P}_i . Any filtered colimit is taken in $[\mathbb{R}, \text{Vect}_\Lambda]$ under Appendix A’s scope policy and is returned to $\text{Pers}_\Lambda^{\text{cons}}$ only by verified constructibility on the window, or by applying \mathbf{T}_τ (and logging the policy in `run.yaml`).

N.2. Projection formula / base change at the persistence layer

Let $f : X \rightarrow Y$ and a Cartesian square

$$\begin{array}{ccc} X' & \xrightarrow{g'} & X \\ f' \downarrow & & \downarrow f \\ Y' & \xrightarrow{g} & Y \end{array}$$

satisfy (N0)–(N2). For filtered complexes F on X and G on Y , write $\mathcal{R}(F), \mathcal{R}(G)$ for their realizations in D_c^b , computed objectwise in t .

Theorem K.2 (PF/BC transported to \mathbf{P}_i and \mathbf{T}_τ [Spec]). *Under (N0)–(N4) the following canonical isomorphisms hold in $\text{Pers}_\Lambda^{\text{cons}}$, natural in (i, τ, f, g, F, G) , and are asserted after truncation by \mathbf{T}_τ (canon order):*

$$(PF) \quad \mathbf{T}_\tau \mathbf{P}_i \left(Rf_* (\mathcal{R}(F) \otimes^L f^* \mathcal{R}(G)) \right) \cong \mathbf{T}_\tau \mathbf{P}_i \left(Rf_* \mathcal{R}(F) \otimes^L \mathcal{R}(G) \right),$$

$$(BC) \quad \mathbf{T}_\tau \mathbf{P}_i \left(Lg^* Rf_* \mathcal{R}(F) \right) \cong \mathbf{T}_\tau \mathbf{P}_i \left(Rf'_* Lg'^* \mathcal{R}(F) \right).$$

Proof sketch. PF and BC are canonical isomorphisms in D_c^b under (N0)–(N2). Evaluate objectwise in t (N3), identify \mathbf{P}_i with $(-i)$ -cohomology in t , then apply \mathbf{T}_τ . Exactness of \mathbf{T}_τ (Appendix A) preserves short exact sequences and hence preserves isomorphisms after passing to \mathbf{P}_i . Naturality in (f, g, F, G) follows from naturality of PF/BC in the six-functor formalism; naturality in (i, τ) follows from functoriality of \mathbf{P}_i and \mathbf{T}_τ . \square

Declaration K.3 (PF/BC after collapse (non-negotiable canon)). Projection formula and base change are audited and asserted at the persistence layer *only after* applying \mathbf{T}_τ (equivalently, after C_τ on the filtered side, then \mathbf{P}_i , then \mathbf{T}_τ). Any observed post-truncation discrepancy is treated as an implementation/hypothesis violation and is logged as budget terms δ^{disc} and δ^{meas} (never as an “algebraic” PF/BC defect).

Corollary K.4 (Compatibility with a filtered lift C_τ [Spec]). *Assume in addition that C_τ is a filtered lift of \mathbf{T}_τ in the sense of Appendix B: for each i , $\mathbf{P}_i(C_\tau F) \simeq \mathbf{T}_\tau \mathbf{P}_i(F)$ up to f.q.i. on the filtered side and equality in $\text{Pers}_\Lambda^{\text{cons}}$ after applying \mathbf{T}_τ . Then the PF/BC isomorphisms of Theorem K.2 may be expressed using $\mathcal{R}(C_\tau F)$ and $\mathcal{R}(C_\tau G)$ on the filtered-realization side, provided comparisons still follow the canon order $\mathbf{P}_i \rightarrow \mathbf{T}_\tau$.*

Remark K.5 (What is *not* claimed). No global Lipschitz control for PF/BC is asserted beyond: (i) the 1-Lipschitz property of \mathbf{T}_τ (Appendix A), and (ii) any additional commutation controls explicitly ledgered (Appendix L). PF/BC are exact identities at the sheaf layer; any persistence-level drift *after truncation* indicates violated hypotheses, window mismatch, or implementation drift and must be logged (Section N.3).

N.3. Windowed protocol and reproducible audit

All PF/BC audits are *windowed*. The mandatory comparison order is:

$$\boxed{\text{for each } t \implies \text{apply } \mathbf{P}_i \implies \text{apply } \mathbf{T}_\tau \implies \text{compare in } \text{Pers}_\Lambda^{\text{cons}}}.$$

Use the *same* MECE, right-open windows and the *same* τ as the rest of the run (Appendix G).

Declaration K.6 (Audit checklist (per window; [Spec])). Record in `run.yaml`: (i) the PF/BC hypothesis set used (proper/smooth, finite Tor, degree normalization and any $f^!$ shift policy); (ii) the functors and objects compared (e.g. $Rf_*(\mathcal{R}(F) \otimes^L f^* \mathcal{R}(G))$ vs. $Rf_* \mathcal{R}(F) \otimes^L \mathcal{R}(G)$); (iii) the verdict (passed:true/false) and, if any post-truncation drift is observed, its breakdown into δ^{disc} and δ^{meas} together with tolerance(s); (iv) the window id and τ used for the comparison, matching those used by B-Gate⁺ and the δ -ledger (Appendix G).

Remark K.7 (Definable coverage). If windows and filtrations are definable in an o-minimal structure, event counts on bounded windows are finite (Appendix H), coverage is decidable, and PF/BC checks reduce to finitely many window-wise persistence comparisons recorded in `run.yaml`.

N.4. Budget integration and window pasting

When the hypotheses hold and comparisons follow the canon order, PF/BC contribute $\delta^{\text{alg}} = 0$ (algebraic defect zero). Only discretization/measurement residuals are budgeted.

Definition K.8 (PF/BC residual defect (windowed)). On a window W and degree i , define the PF and BC observed residuals (post-truncation) by

$$\begin{aligned}\delta_W^{\text{PF}}(i; \tau) &:= d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(Rf_*(\mathcal{R}(F) \otimes^L f^* \mathcal{R}(G))), \mathbf{T}_\tau \mathbf{P}_i(Rf_* \mathcal{R}(F) \otimes^L \mathcal{R}(G))\right), \\ \delta_W^{\text{BC}}(i; \tau) &:= d_{\text{int}}\left(\mathbf{T}_\tau \mathbf{P}_i(Lg^* Rf_* \mathcal{R}(F)), \mathbf{T}_\tau \mathbf{P}_i(Rf'_* Lg'^* \mathcal{R}(F))\right).\end{aligned}$$

These are expected to be 0 in ideal algebraic settings; any nonzero value is recorded as $\delta^{\text{disc}} + \delta^{\text{meas}}$ (and optionally split further by implementation source).

Proposition K.9 (Pipeline budget integration). *Fix a commutative quantale budget law $(V, \oplus, \leq, 0)$ (Appendix K–L) and a window W . In a pipeline that includes Mirror/Transfer commutation bounds and A/B residuals (Appendix L/K), the window budget in degree i is*

$$\Sigma \delta_W(i) = \left(\widehat{\oplus}_{\text{Mirror/Transfer}} \delta^{(j)}(i, \tau_j) \right) \widehat{\oplus} \left(\widehat{\oplus}_{A/B \text{ fails}} \delta^{(\text{AB})} \right) \widehat{\oplus} \delta_W^{(\text{PF/BC})}(i),$$

where $\delta_W^{(\text{PF/BC})}(i)$ has only the PF/BC coordinate(s) nonzero, equal to the recorded residuals (Definition K.8) under the chosen split. Collapsing once by the declared homomorphism $\pi : V^{\times m} \rightarrow V$ yields the scalar $\Sigma \delta_W(i)$, compatible with Appendix L and Appendix K. This scalar is nonincreasing under any subsequent 1-Lipschitz persistence post-processing.

Corollary K.10 (Window pasting (Restart/Summability alignment)). *Over a MECE partition $\{W_k\}$, the sum (quantale aggregation) of $\Sigma \delta_{W_k}(i)$ controls end-to-end discrepancy. Compare $\Sigma \delta_{W_k}(i)$ against the per-window safety margins $\text{gap}_{\tau_k}(i)$ to apply Restart and Summability (Appendix J), enforcing $\text{gap}_{\tau_k}(i) > \Sigma \delta_{W_k}(i)$ per window.*

N.5. Ext–tests under change of functor / coefficients

PF/BC isomorphisms transport Ext^1 -tests along canonical identifications.

Proposition K.11 (Portability of the Ext^1 -test (sheaf layer)). *Under (N0)–(N2), any isomorphism $A \xrightarrow{\sim} B$ in D_c^b induces a natural isomorphism*

$$\text{Ext}^1(A, \Lambda) \xrightarrow{\sim} \text{Ext}^1(B, \Lambda).$$

In particular, if $\text{Ext}^1(\mathcal{R}(C_\tau F), \Lambda) = 0$, then Ext^1 also vanishes for any PF/BC partner of $\mathcal{R}(C_\tau F)$ (under the same hypothesis regime).

Remark K.12 (Bridge stays one-way). The one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ (Appendix C) is unchanged. PF/BC only *transport* the Ext^1 test across equivalent sheaf-theoretic descriptions; no converse implication and no new equivalence is claimed.

N.6. Functoriality and two–out–of–three (windowed Beck–Chevalley)

Proposition K.13 ([Spec] Two–out–of–three for PF/BC squares (after collapse)). *Fix a Cartesian square, a window W , and $\tau > 0$. If any two among the PF/BC isomorphisms (evaluated objectwise in t , then passed through \mathbf{P}_i , then truncated by \mathbf{T}_τ) hold as isomorphisms in $\text{Pers}_\Lambda^{\text{cons}}$, then the third holds as well (all comparisons performed on W and in degree i).*

Proof sketch. Two–out–of–three holds in D_c^b for composable isomorphisms arising from PF/BC. Objectwise evaluation and exactness of \mathbf{T}_τ transport this to $\text{Pers}_\Lambda^{\text{cons}}$ after applying the canon order. \square

N.7. Implementation notes and checkpoints

- **Finite windows / constructibility.** On bounded t -windows, bar events are finite (Appendix H). PF/BC are computed objectwise in t and the resulting persistence objects remain constructible on windows; if not, enforce return via \mathbf{T}_τ and log the policy.
- **Exactness bookkeeping (allowed primitives).** Reductions to persistence use only: (i) PF/BC hold in D_c^b under (N2); (ii) \mathbf{P}_i reads $(-i)$ -cohomology under (N3); (iii) \mathbf{T}_τ is exact and 1-Lipschitz (Appendix A); (iv) filtered colimits respect the scope policy (Appendix A, Remark .67).
- **Proper/smooth reminder.** We use cohomological conventions; any $f^!$ -induced shifts in smooth variants are absorbed by (N3) and must be logged (degree normalization clause) if they affect comparisons.
- **Window coherence (non-negotiable).** PF/BC audits must use the *same* windows and τ as B-Gate⁺ and the δ -ledger (Appendix G); otherwise budget accounting and Restart/Summability pasting (Appendix J) become invalid.

N.8. Formalization stubs (Lean/Coq) [Spec]

A minimal API (cf. Appendix F) includes:

- `pf_iso`: $Rf_*(A \otimes f^*B) \cong Rf_*A \otimes B$ under properness and Tor-finiteness;
- `bc_iso`: $Lg^*Rf_*A \cong Rf'_*Lg'^*A$ for Cartesian squares (and smooth variants with $f^!$ and the shift bookkeeping of (N3));
- `eval_t`: objectwise evaluation in t ;
- `to_pers`: extraction of \mathbf{P}_i and truncation by \mathbf{T}_τ (canon order);
- `pfbc_pers_nat`: naturality in (i, τ, f, g, F, G) after collapse;
- `budget_hook`: logging of post-truncation residuals as $\delta^{\text{disc}}, \delta^{\text{meas}}$ and insertion into the (product) ledger aligned with Appendix G and Appendix L/K.

Summary. Under standard PF/BC hypotheses over a field Λ , projection formula and base change descend—via objectwise evaluation in t , \mathbf{P}_i , and exact truncation \mathbf{T}_τ —to canonical, natural isomorphisms at the persistence layer. All comparisons follow the windowed protocol “for each $t \rightarrow \text{persistence} \rightarrow \text{collapse} \rightarrow \text{compare}$,” using the same MECE right-open windows and the same τ as the rest of the run. PF/BC contribute zero algebraic defect; any post-truncation drift is treated as discretization/measurement residual and accounted for in the δ -ledger, integrated with Mirror/Transfer commutation (Appendix L), multi-axis reflectors (Appendix K), and Restart/Summability pasting (Appendix J), while keeping the one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ intact (Appendix C).

Appendix O. Fukaya Realization & Stability [Spec + Permitted Ops + δ -Ledger + B-Gate⁺] (Reinforced, canon-aligned)

Standing conventions (canon). We work over a coefficient *field* Λ (e.g. a ground field k or a Novikov field). All persistence modules are constructible; all equalities asserted in this appendix are asserted at the persistence layer in the one-parameter constructible setting, and all audits are performed *after collapse* on the B-side single layer. Filtered (co)limits are computed objectwise in $[\mathbb{R}, \text{Vect}_\Lambda]$ under the scope policy of Appendix A, Remark .67, and then (when stated) returned to the constructible range (by verified constructibility

on the window or by applying \mathbf{T}_τ). Distances are measured by the interleaving metric d_{int} (= bottleneck in the constructible 1D setting). Truncation \mathbf{T}_τ is the exact bar-deletion reflector and is 1-Lipschitz (Appendix A, Proposition .70); chain models are used only up to filtered quasi-isomorphism (f.q.i.) in $\text{Ho}(\text{FiltCh}(\Lambda))$. Global conventions: Ext^1 is always taken against $\Lambda[0]$; the energy exponent satisfies $\alpha > 0$ (default $\alpha = 1$). References to “generic fiber dimension / infinite bars” point to Appendix D, Remark A.2. Monotonicity claims follow the global policy: *deletion-type only* (nonincreasing), inclusion-type merely stable/non-expansive (Appendix E). Windows are MECE and right-open; bars are half-open $[b, d)$ (Appendix G/H). All comparisons follow the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare}}$$

after collapse on the B-side single layer (Appendix G).

Remark L.1 (Right-open windows and half-open bars). Right-open windows and half-open $[b, d)$ bars enforce MECE coverage and avoid double-counting at endpoints; events at a right boundary are attributed to the subsequent window.

Definition L.2 (Filtered chain model and persistence). A filtered chain complex over Λ means a chain complex C_\bullet equipped with an exhaustive, increasing filtration $\{F^t C_\bullet\}_{t \in \mathbb{R}}$ by subcomplexes such that the differential preserves the filtration and continuation data act by filtered maps. For each homological degree i , the degree- i persistence module is $t \mapsto H_i(F^t C_\bullet) \in \text{Vect}_\Lambda$. Constructibility on bounded windows means only finitely many jumps (break times) occur and ranks are finite on bounded intervals.

Definition L.3 (Deletion-type operation at the persistence layer). A morphism $M \rightarrow N$ in Pers_Λ is *deletion-type* on a window $W \subset \mathbb{R}$ if, after restricting to W and post-composing with \mathbf{T}_τ (for any $\tau \geq 0$), the induced effect can only shorten or remove existing bars and cannot create new bars on W . Equivalently, all deletion-type monotone indicators of Appendix E are nonincreasing under this operation (and inclusion-type indicators are not claimed monotone).

O.1. Realization functor and hypotheses

[Spec] Fix a Liouville/Weinstein sector (X, λ) with a (possibly empty) system of stops. Write $\text{Fuk}(X; \text{stops})$ for a wrapped/exact/monotone Fukaya-type category for which Floer-theoretic chain models admit an *action filtration*. We normalize the filtration parameter as follows: *the action value is the filtration parameter t* , increasing with larger action (so sublevel sets F^t mean action $\leq t$). Package the chain-level construction into a realization

$$\mathcal{F} : (\text{geometric input}) \longrightarrow \text{FiltCh}(\Lambda),$$

natural in continuation data and stop operations, with degree- i persistence $\mathbf{P}_i(\mathcal{F}(-)) \in \text{Pers}_\Lambda^{\text{cons}}$ (constructible on bounded windows). We assume:

- (O0) **Coefficients/admissibility.** Λ is a field; in the monotone/exact regimes with admissible almost complex structures, the action and index filtrations are well defined; continuation solutions have finite energy.
- (O1) **Constructibility on bounded action windows.** On every bounded action window $[0, \sigma]$ the Floer complexes have finitely many generators and finitely many break times; hence $\mathbf{P}_i(\mathcal{F}(-))$ is constructible on $[0, \sigma]$. The same local finiteness holds with Novikov coefficients on bounded windows.
- (O2) **Continuation shift bound.** Any continuation map for a homotopy of data with controlled action shift ε induces a filtered chain map whose filtration increase is $\leq \varepsilon$.

- (O3) **Stop operations are deletion-type (post-collapse).** Adding a stop or shrinking a sector removes generators and/or increases differentials in a way that corresponds to a deletion-type operation at the persistence layer: on any fixed action window, no new bars are created *after applying* \mathbf{T}_τ .
- (O4) **Up to filtered quasi-isomorphism.** Chain models are considered up to f.q.i.; all claims are invariant under f.q.i. and are asserted after passing to persistence and truncating by \mathbf{T}_τ .

Remark L.4 (Action filtration normalization). Our choice “action value equals filtration parameter” fixes the direction of filtration. Monotone time reparametrizations that preserve order act by reindexings and, after normalization, give isometries in d_{int} at the persistence layer.

O.2. Stability: continuation and stops

Theorem L.5 (Continuation control implies interleaving bound). *Under (O2), for any two realizations related by a continuation with action shift ε ,*

$$d_{\text{int}}(\mathbf{P}_i(\mathcal{F}_0), \mathbf{P}_i(\mathcal{F}_1)) \leq \varepsilon, \quad d_{\text{int}}(\mathbf{T}_\tau \mathbf{P}_i(\mathcal{F}_0), \mathbf{T}_\tau \mathbf{P}_i(\mathcal{F}_1)) \leq \varepsilon$$

for all i and all $\tau \geq 0$.

Proposition L.6 (Deletion-type monotonicity for stops (post-collapse)). *Under (O3), adding a stop or shrinking a sector induces, on any window and after \mathbf{T}_τ , a deletion-type morphism: for every i and $\tau \geq 0$,*

$$\mathbf{T}_\tau \mathbf{P}_i(\mathcal{F}_{\text{with stop}}) \preceq \mathbf{T}_\tau \mathbf{P}_i(\mathcal{F}_{\text{without stop}}),$$

and all deletion-type monotone indicators of Appendix E are nonincreasing under this operation.

Theorem L.7 (Stop/continuation policy — canon-aligned). *In the action-filtered Fukaya realization, stop addition is deletion-type (post-collapse nonincrease) and ε -continuation maps satisfy $d_{\text{int}} \leq \varepsilon$ on \mathbf{T}_τ -collapsed persistence. In the δ -ledger: continuation/shift steps record $\delta^{\text{alg}} = \varepsilon$; deletion-type stop steps record $\delta^{\text{alg}} = 0$.*

Declaration L.8 (Gate Cascade placement (after-collapse, B-side single layer)). Fukaya realizations enter the Gate Cascade at the *B-side* after applying \mathbf{P}_i and truncating by \mathbf{T}_τ , and before any Mirror/Tropical post-processing (Appendix L) or multi-axis reflector interactions (Appendix K). All audits and budgets for Fukaya steps therefore use the order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{Fukaya-op compare}},$$

with the same window and τ as B-Gate⁺ (Appendix J/G).

Remark L.9 (Budget-adjusted continuation radius). On a window W with additive δ -ledger (Appendix L/K), use the effective radius $\varepsilon_{\text{eff}} := \varepsilon + \Delta_W$ (Appendix I) when invoking survival or matching claims. Here Δ_W aggregates only *non-Fukaya* defects (e.g. Mirror/Transfer commutation bounds, A/B residuals, discretization/measurement slack).

O.3. Towers, comparison map, and diagnostics

Let $F = (F_n)_{n \in I}$ be a directed system of geometric inputs (e.g. refining Hamiltonians/perturbations or nested stop systems) with colimit F_∞ . Apply \mathcal{F} and then \mathbf{P}_i to obtain a tower in $\text{Pers}_\Lambda^{\text{cons}}$. For $\tau \geq 0$ consider the comparison map (Appendix J), always formed *after collapse*:

$$\phi_{i,\tau}(F) : \lim_{\longrightarrow n} \mathbf{T}_\tau(\mathbf{P}_i(\mathcal{F}(F_n))) \longrightarrow \mathbf{T}_\tau(\mathbf{P}_i(\mathcal{F}(F_\infty))).$$

Definition L.10 (Sufficient tower hypotheses (canon references)). Assume any one of the sufficient regimes from Appendix D, § D.3 (e.g.):

- (S1) truncation/colimit commutation up to vanishing error under the scope policy, after applying \mathbf{T}_τ ;
- (S2) no τ -accumulation of break times on the window (Type IV exclusion at the chosen τ);
- (S3) a Cauchy condition in d_{int} with compatible structure maps (post-collapse).

Each regime ensures stability of barcodes under the colimit and truncation at the persistence layer.

Theorem L.11 (When the comparison is an isomorphism). *Assume a continuation-controlled tower: there exist bounds $\varepsilon_n \rightarrow 0$ such that*

$$d_{\text{int}}(\mathbf{P}_i(\mathcal{F}(F_n)), \mathbf{P}_i(\mathcal{F}(F_\infty))) \leq \varepsilon_n,$$

and assume any of (S1)/(S2)/(S3) from Definition L.10. Then $\phi_{i,\tau}(F)$ is an isomorphism for all $\tau \geq 0$. Consequently the tower diagnostics vanish:

$$\mu_{i,\tau}(F) = u_{i,\tau}(F) = 0,$$

where (μ, u) are the generic fiber dimensions of $\ker(\phi_{i,\tau})$ and $\text{coker}(\phi_{i,\tau})$ (Appendix D, Remark A.2).

Corollary L.12 (Grid \Rightarrow continuum survival (budget-aware)). *In discretization towers (mesh $h \rightarrow 0$) with certified continuation bounds $\varepsilon(h) \rightarrow 0$, any bar detected in a fixed window $[0, \tau_0]$ whose \mathbf{T}_{τ_0} -clipped length exceeds $2\varepsilon_{\text{eff}}(h)$ survives in the limit, where $\varepsilon_{\text{eff}}(h) = \varepsilon(h) + \Delta_W$ (Appendix I).*

O.4. Permitted-operations table (windowed, post-collapse) and δ -ledger

All comparisons follow the protocol “for each $t \rightarrow$ persistence $\mathbf{P}_i \rightarrow$ collapse $\mathbf{T}_\tau \rightarrow$ compare,” on MECE, right–open windows and a fixed τ (Appendix G/N). The table summarizes permitted operations, their type, quantitative contracts (post-collapse), and how to record them in the δ -ledger.

Operation	Type	Quantitative contract after \mathbf{T}_τ and ledger entry
Add stop / shrink sector	Deletion	Deletion-type: no new bars on the window after \mathbf{T}_τ ; all deletion indicators nonincreasing (Appendix E). Ledger: $\delta^{\text{alg}} = 0$; record $\delta^{\text{disc}}, \delta^{\text{meas}}$ if any.
Continuation (tame homotopy)	Shift	$d_{\text{int}} \leq \varepsilon$ (Theorem L.5). Ledger: $\delta^{\text{alg}} = \varepsilon$.
Hamiltonian change (bounded drift)	Shift	Reduce to continuation control: $d_{\text{int}} \leq \varepsilon$ post-collapse. Ledger: $\delta^{\text{alg}} = \varepsilon$.
Almost complex structure change (tame)	Shift	Reduce to continuation control: $d_{\text{int}} \leq \varepsilon$ post-collapse. Ledger: $\delta^{\text{alg}} = \varepsilon$.
Regrading / Maslov shift	Bookkeeping	Degree reindexing isometric after normalization (no metric cost). Ledger: $\delta^{\text{alg}} = 0$.
Monotone time reparametrization	Reindex	Isometry after reindex normalization (Appendix G policy). Ledger: $\delta^{\text{alg}} = 0$.
Mirror/Tropical/Transfer processing	post- External	Accounted by Appendix L: if commutation defect $\delta(i, \tau)$ is used, ledger it in the appropriate channel (product ledger) and collapse once by π .
Non-nested reflectors (if used)	External	A/B test or soft-commuting fallback (Appendix K/L); ledger: $\delta^{\text{alg}} = \Delta_{\text{comm}}$ when fallback residual is incurred.

Definition L.13 (δ -ledger and budgets (windowed, post-collapse)). Per window $W = [u, u')$ and degree i , define the aggregate budget as a commutative quantale sum (Appendix K/L):

$$\Sigma\delta_W(i) := \sum_{\text{continuations/shifts}} \varepsilon + \sum_{\text{Mirror/Transfer}} \delta(i, \tau) + \sum_{\text{A/B fails}} \Delta_{\text{comm}} + \sum_{\text{audits}} (\delta^{\text{disc}} + \delta^{\text{meas}}),$$

where “audits” include PF/BC checks (Appendix N), numerical tolerances, and any scope-policy return-to-constructible enforcement costs (Appendix A/G). All terms are measured/recorded *after* applying \mathbf{T}_τ and on the same MECE right-open windowing.

Remark L.14 (Product-ledger compatibility). If the run uses a product quantale ledger (Appendix L, §L.3), record Fukaya continuation terms in the dedicated channel (e.g. `fukaya_shift`) and collapse once via π . This prevents double counting across external channels (mirror/transfer/A/B/audits).

O.5. B-Gate⁺, restart, and summability (window pasting)

We adopt B-Gate⁺ with a per-window safety margin $\text{gap}_\tau(i) > 0$ computed after \mathbf{T}_τ . On window W and degree i , the gate *passes* if

$$\text{gap}_\tau(i) > \Sigma\delta_W(i).$$

Across consecutive windows $(W_k)_k$, assume: (i) transitions are finite compositions of deletion-type steps and ε -continuations measured post-collapse, and (ii) Summability holds $\sum_k \Sigma\delta_{W_k}(i) < \infty$ (Appendix J). Then the Restart inequality of Appendix J yields for some $\kappa \in (0, 1]$,

$$\text{gap}_{\tau_{k+1}}(i) \geq \kappa \left(\text{gap}_{\tau_k}(i) - \Sigma\delta_{W_k}(i) \right),$$

so positivity of the margin persists along the pipeline. Per-window certificates paste to a global certificate on $\bigcup_k W_k$ (Appendix J, Theorem J:G.20).

Remark L.15 (Choice of κ). The constant κ accounts for uniform losses at window transitions (e.g. finite alignment overheads or reindex coercions). Under exact commutation and perfectly aligned window policies one may take $\kappa = 1$.

O.6. Windowed usage and run.yaml alignment

Record in `run.yaml` per window and degree:

- the operation sequence (stops/sector changes, continuations) with quantitative parameters (ε , thresholds τ , sweep settings);
- the δ -ledger entries (product vector if used) and the collapsed scalar $\Sigma\delta_W(i)$;
- the B-Gate⁺ safety margin $\text{gap}_\tau(i)$ and pass/fail verdict;
- any external steps (Mirror/Tropical/Transfer, reflectors) with A/B policy data (η , Δ_{comm}) (Appendix K/L);
- constructibility checks: bounds on generators and event counts on the window (Appendix H) and any enforced return-to-constructible via \mathbf{T}_τ (Appendix A).

All diagnostics (μ, u) and comparison maps $\phi_{i,\tau}$ are computed *after* truncation \mathbf{T}_τ and logged with the same window and τ .

O.7. Failure modes and audit checklist

Failure modes (outside our scope).

- **Loss of filtration control.** Non-admissible data or bubbling may invalidate (O2); quantitative continuation bounds then fail and must not be asserted.
- **Near-threshold accumulation (Type IV).** Accumulation of bar lengths near τ can break stabilization and invalidate tower claims without (S2)/(S3) (Appendix D).
- **Inclusion-type operations.** Removing stops or enlarging sectors can create new features; monotonicity is not claimed. Only stability via continuation control may be used, and only post-collapse.

Audit checklist (runtime verifications).

1. Record continuation shift bounds ε and certify $d_{\text{int}} \leq \varepsilon$ post-collapse (Theorem L.5); ledger $\delta^{\text{alg}} = \varepsilon$.
2. Verify constructibility on each window (finite generators/events; Appendix H) and log per-window counts (Appendix G).
3. For stop additions/sector shrinkage, mark operation as deletion-type and evaluate Appendix E deletion indicators after \mathbf{T}_τ .
4. For towers, log ε_n , verify one of (S1)/(S2)/(S3), compute $(\mu, u); \phi_{i,\tau} \text{ iso} \Rightarrow (\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ (Appendix D/J).
5. If external functors are used, run Mirror/Transfer commutation audits and A/B tests (Appendix L/K) and incorporate residuals into $\Sigma\delta_W(i)$ using the declared ledger law.

O.8. Formalization stubs (Lean/Coq) [Spec]

A minimal API (cf. Appendix F) includes:

- `fukaya_realize`: returns an action-filtered chain model \mathcal{F} up to f.q.i., with constructibility on bounded windows (O1).
- `cont_eps`: continuation maps with filtration increase $\leq \varepsilon$ implying $d_{\text{int}} \leq \varepsilon$ after truncation (Theorem L.5).
- `stop_delete`: deletion-type morphisms for stop addition/sector shrink post-collapse (Proposition L.6).
- `tower_phi_iso`: sufficient criteria ensuring $\phi_{i,\tau}$ is an isomorphism and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ (Appendix D/J).
- Budget hooks: `ledger_add`, `gate_plus`, and restart/summability contracts aligned with Appendix J and Appendix L/K product-ledger semantics.

O.9. Examples and regression tests

Example L.16 (Exact wrapped setting with a new stop). Let $X = T^*Q$ with its standard exact form and consider a wrapped setting with a stop at infinity. Adding an additional stop supported on a Legendrian subset removes Reeb chords crossing the stop. On any fixed action window, no new generators appear and differentials can only increase; hence the induced operation is deletion-type post-collapse (Proposition L.6). Regression test: after \mathbf{T}_τ , deletion indicators (Appendix E) weakly decrease and no new bars appear on the window.

Example L.17 (Continuation bound from Hamiltonian drift). Suppose H_0, H_1 are cofinal Hamiltonians with $\sup(H_1 - H_0) \leq \varepsilon$ on the relevant support and homotoped by a tame path. The action change along continuation solutions is bounded by ε ; thus $d_{\text{int}} \leq \varepsilon$ post-collapse (Theorem L.5). Regression test: barcode bottleneck distance between the two collapsed persistence outputs never exceeds the certified ε .

Example L.18 (Grid-to-continuum). Discretize a time-dependent Floer datum with mesh h and certified continuation bound $\varepsilon(h) \rightarrow 0$. For any fixed τ_0 , bars of \mathbf{T}_{τ_0} -collapsed persistence with clipped length $> 2\varepsilon_{\text{eff}}(h)$ survive in the limit (Corollary L.12). Regression test: survival rates converge after accounting for the per-window $\Sigma\delta_W(i)$ and the B-Gate⁺ margin condition.

O.10. Summary

Floer-theoretic realizations with action filtration yield constructible persistence on bounded windows (O1). Continuation with shift ε implies $d_{\text{int}} \leq \varepsilon$ at the persistence layer and remains valid after truncation (Theorem L.5). Adding stops or shrinking sectors is deletion-type and hence nonincreasing for all deletion indicators after collapse (Proposition L.6). *Placement*: Fukaya steps are audited *after collapse* on the B-side single layer (Declaration L.8); the same window/ τ are used for B-Gate⁺, Mirror/Transfer, and PF/BC checks. A windowed permitted-ops table prescribes how to assign δ -ledger entries and how to integrate external commutation/A-B residuals without double counting (Appendix L/K). B-Gate⁺ requires $\text{gap}_\tau > \Sigma\delta$ per window and pastes via Restart/Summability (Appendix J). Under standard tower hypotheses (Appendix D), the comparison map $\phi_{i,\tau}$ is an isomorphism and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ (Theorem L.11); budget-aware grid-to-continuum survival follows (Appendix I). All items respect MECE right-open windows, half-open bars, and the canon order “ $t \rightarrow \mathbf{P}_i \rightarrow \mathbf{T}_\tau \rightarrow \text{compare}$,” and integrate with the reproducible run.yaml workflow (Appendix G).

Appendix P. Tropical–LMHS Dictionary [Spec; after-collapse indicators only] (canon-aligned)

Standing conventions (canon). We work over a coefficient field k . All persistence modules are constructible on bounded windows; filtered (co)limits are computed objectwise in $[\mathbb{R}, \text{Vect}_k]$ under the scope policy of Appendix A and returned to the constructible range when stated (by verified constructibility on the window or by applying the exact bar-deletion reflector \mathbf{T}_τ). *All numeric comparisons in this appendix are evaluated after collapse* in the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare on } \mathbf{T}_\tau \mathbf{P}_i},$$

with the same MECE, right-open windows and the same τ as elsewhere (Appendix G/N). Distances/defects aggregate in a declared commutative *quantale* V with sum \oplus , order \preceq , and scalar action \odot (Appendix K/L/S). The interleaving metric d_{int} (= bottleneck in the constructible 1D setting) is used on $\text{Pers}_k^{\text{cons}}$, and \mathbf{T}_τ is exact and 1-Lipschitz (Appendix A). Any chain-level operation C_τ is used only up to f.q.i., and only through its induced persistence after applying \mathbf{P}_i and \mathbf{T}_τ (Appendix B). The one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ is used only in $D^b(k\text{-mod})$ and only forward (Chapter 3; Appendix C). No converse and no new global equivalences are asserted.

P.1. Dictionary contract (Spec) and safe use

Declaration M.1 (Tropical–LMHS dictionary (Spec; advisory, after-collapse only)). On a definable window W (Appendix H/J), a lookup procedure maps tropical inputs (val, trop_λ , fan data) to coarse LMHS proxies

$(W_\bullet, N, h_\infty^{p,q})$. These proxies are used *only* to propose *after-collapse* indicators on $\mathbf{T}_\tau \mathbf{P}_i(F|_W)$ for selected degrees i . The dictionary is *never* a Gate and *never* a certificate: it may *trigger* certified tests (Appendix J/N/L), but cannot certify them.

- Remark M.2** (Safe-use policy). 1. **Advisory only.** Dictionary proposals are advisory and may contribute at most an *advisory* budget entry $\delta^{\text{spec}} \in V$ (Definition M.7). If no evidence supports calibration, set $\omega(W) = 0$ so $\delta^{\text{spec}} = 0$.
2. **Certified decisions.** Any decision impacting PF/BC, Overlap Glue, tower diagnostics (μ, u) , or classification must rely only on certified after-collapse statements (Appendix N/J/L/D) and the declared window protocol (Appendix G).
3. **Finite verification cost.** All claims are windowed and reproducible; on bounded definable windows, event counts are finite and Čech depth is finite (Appendix H/J), hence all required checks terminate.

P.2. Lookup targets and after-collapse indicators

Tropical input	LMHS proxy (coarse)	After-collapse indicator (advisory; non-gating)
val(moduli) step loci	weight jumps in W_\bullet	candidate deletion change-points (stop/shrink <i>hints</i>) to be verified as deletion-type post-collapse
trop $_\lambda$ slopes ($\lambda \rightarrow 0^+$)	rk $N = \text{rk log } T_u$	suggested local bins / expected smallness of windowed E_1 (trigger certified tests)
Balanced-fan combinatorics	split vs. non-split type	additivity expectations for tower diagnostics (μ, u) across overlaps (to be confirmed by Overlap Glue)
Tropical periods/lengths	limiting $h_\infty^{p,q}$ pattern	energy-bin prioritization; ε -survival <i>hints</i> subject to budget-adjusted radii (Appendix I/L)

Remark M.3 (Definable coverage). Windows W are finite unions of half-open intervals definable in a fixed o-minimal structure; hence finite event decomposition and finite Čech depth hold, yielding finite verification cost for all local tests and overlap checks (Appendix H/J).

P.3. Quantitative commutation and 2-cell bounds (after collapse)

Declaration M.4 (Tropical 2-cell bound (after-collapse)). Let trop $_\lambda$ be a Mirror/Tropical post-processing step that is 1-Lipschitz at the persistence layer and admits a controlled 2-cell with collapse in the sense of Appendix L. On a window W and for degree i , record a bound $\delta_{\text{trop}}(i, \tau; W) \in V$ such that

$$d_V(\mathbf{T}_\tau \mathbf{P}_i(\text{trop}_\lambda C_\tau F), \mathbf{T}_\tau \mathbf{P}_i(C_\tau \text{trop}_\lambda F)) \preceq \delta_{\text{trop}}(i, \tau; W).$$

Enter $\delta_{\text{trop}}(i, \tau; W)$ into the window budget (Definition M.11) using the declared quantale sum \oplus .

Remark M.5 (non-expansive measurement). All discrepancies are measured after \mathbf{T}_τ . Therefore any contraction in trop $_\lambda$ is preserved, and subsequent 1-Lipschitz persistence post-processing cannot increase recorded bounds (Appendix L).

Remark M.6 (Product-ledger compatibility). If the run uses a product quantale ledger (Appendix L, §L.3), record $\delta_{\text{trop}}(i, \tau; W)$ in the dedicated channel (e.g. tropical_2cell) and collapse once via π . This avoids double counting across channels.

P.4. Confidence-weighted advisory defect δ^{spec}

Definition M.7 (Confidence weight and advisory defect). Each dictionary proposal on W carries a confidence weight $\omega(W) \in [0, 1]$ (dimensionless). Given a raw advisory magnitude $\widehat{\delta}_{\text{spec}}(i, \tau; W) \in V$ (e.g. from suggested bin gaps, slope tolerances, or disagreement between proxy and measured after-collapse indicators), define the ledgered advisory term

$$\delta^{\text{spec}}(i, \tau; W) := \omega(W) \odot \widehat{\delta}_{\text{spec}}(i, \tau; W) \in V.$$

Calibration of $\omega(W)$ may use held-out windows, stability-band cross-checks, or τ -sweeps (Appendix J/M). Absent evidence, set $\omega(W) = 0$ (no advisory cost).

Remark M.8 (Summability on countable covers). For a locally finite countable definable cover $\{W_j\}$ of a bounded window, $\bigoplus_j \delta^{\text{spec}}(i, \tau; W_j)$ is well-defined whenever the corresponding series converges in V . This is the *T-Delta-Sum-Converges* condition, compatible with Restart/Summability (Appendix J).

P.5. Local trigger via E_1 (after-collapse)

Windowed E_1 indicator (after collapse). Fix a window $W = [u, u']$. For a degree i , define the windowed E_1 indicator on W by

$$E_1^{(i)}(F; W, \tau) := \int_u^{u'} \dim_k \left(\mathbf{T}_\tau \mathbf{P}_i(F)(t) \right) dt,$$

i.e. the clipped Betti integral of the post-collapse persistence on W (cf. Appendix M, $\|\cdot\|_{E_1}$). By constructibility, this integral is a finite sum on any bounded window.

Declaration M.9 (Advisory trigger (non-gating)). If the dictionary suggests on W that the monodromy-rank proxy vanishes (no slope/facet change), it may trigger the *certified* test $E_1^{(1)}(F; W, \tau) = 0$ computed on $\mathbf{T}_\tau \mathbf{P}_1(F|_W)$. When verified, one may use the established after-collapse chain of implications in the approved direction:

$$E_1^{(1)}(F; W, \tau) = 0 \implies \mathbf{T}_\tau \mathbf{P}_1(F|_W) = 0 \implies \text{PH}_1(\mathbf{T}_\tau F|_W) = 0 \implies \text{Ext}^1(\mathcal{R}(\mathbf{T}_\tau F)|_W, k) = 0,$$

where the last implication is the one-way bridge used in $D^b(k\text{-mod})$ (Appendix C) and the realization \mathcal{R} is only invoked within its stated hypotheses. Failure of the certified test records a nonzero raw advisory magnitude $\widehat{\delta}_{\text{spec}}$ (e.g. the smallest positive E_1 -bin on the window) and hence a ledgered δ^{spec} via Definition M.7.

Remark M.10 (Overlap Glue with finite depth). On a definable cover of a bounded window, Overlap Glue terminates after finitely many checks (Appendix J). Any residuals are aggregated into the budget by \oplus and never override certified pass/fail decisions.

P.6. Window budget and pasting

Definition M.11 (Window budget with advisory terms). For window W and degree i , define the post-collapse window budget as the quantale sum

$$\Sigma \delta_W(i) := \left(\bigoplus_{\text{Mirror/Tropical 2-cells}} \delta_{\text{trop}}(i, \tau; W) \right) \oplus \left(\bigoplus_{\text{A/B fails}} \Delta_{\text{comm}}(i; W) \right) \oplus \left(\bigoplus_{\text{audits}} (\delta^{\text{disc}} + \delta^{\text{meas}}) \right) \oplus \left(\bigoplus_{\text{advisory}} \delta^{\text{spec}}(i, \tau; W) \right),$$

with all terms measured and recorded after \mathbf{T}_τ and under the same MECE right-open windowing (Appendix G/L/K/N). B-Gate⁺ requires $\text{gap}_\tau(i) > \Sigma \delta_W(i)$ per window (Appendix J).

Remark M.12 (Countable covers and pasting). If a locally finite countable definable cover is used, Summability plus $\bigoplus_W \delta^{\text{spec}}(i, \tau; W)$ convergent ensures pasting across windows (Appendix J). Advisory terms are designed to be optionally zeroed ($\omega = 0$) to preserve summability when needed.

P.7. Minimal working example (non-gating)

Single-facet advisory pass. On W , tropical slopes are constant; the dictionary proposes $N = 0 \Rightarrow E_1^{(1)}(F; W, \tau) = 0$ with $\omega(W) = 0.6$. The certified test returns $E_1^{(1)}(F; W, \tau) = 0$ *true*; no advisory ledger entry is added. If the test fails with smallest nonzero E_1 -bin $\widehat{\delta}_{\text{spec}} = \varepsilon_\star \in V$, then $\delta^{\text{spec}} = 0.6 \odot \varepsilon_\star$ is added to $\Sigma\delta_W(1)$.

P.8. Reproducibility hooks (run.yaml)

Key	Meaning
policy:{after_collapse_only}	Must be true for this appendix's comparisons.
quantale:{name,op,unit,order,scalar}	Quantale V , aggregation \oplus , unit 0_V , order \preceq , scalar action \odot (Appendix K/L/S).
windows:{mece,right_open,intervals}	MECE right-open window considerations and explicit intervals (Appendix G).
definable:{structure>window_formulae}	o-minimal structure and window formulas (Appendix H/J).
tropical:{lambda,contraction_kappa,bins}	Mirror/Tropical configuration and binning policy (Appendix L/M).
lmhs:{proxies}	Enabled proxies: rankN, weights, hpq_inf.
two_cell:{delta_trop}	Recorded $\delta_{\text{trop}}(i, \tau; W)$ bounds per window/degree (Appendix L).
dict:{omega,raw_spec,delta_spec}	Confidence $\omega(W)$, raw $\widehat{\delta}_{\text{spec}}$, and ledgered δ^{spec} .
budget:{sum_delta,channels}	Per-window budget aggregation and (optional) product-ledger channels (Appendix L/K).

P.9. Minimal API (pseudocode) [Spec]

```

def advisory_delta(window, degree, tau, raw_spec, omega):
    # raw_spec in V; omega in [0,1]; uses quantale scalar action  $\odot$ 
    return omega  $\odot$  raw_spec

def window_budget(entries):
    # entries: list of V-elements from tropical 2-cells, A/B residuals,
    # audits (disc/meas), and advisory deltas; aggregate by  $\oplus$ .
    Sigma = 0_V
    for v in entries:
        Sigma = Sigma  $\oplus$  v
    return Sigma

```

P.10. What this appendix does *not* assert (hard boundaries)

No pre-collapse control; no claims of strict commutation beyond recorded 2-cell bounds; no global equivalences beyond the one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ used only in its stated derived setting. The dictionary never acts as a Gate, never overrides certified after-collapse tests, and never upgrades advisory proxies to certificates.

P.11. Summary

A tropical \rightarrow LMHS lookup provides *after-collapse* advisory signals on definable windows. Quantitative effects are integrated through: (i) controlled Mirror/Tropical 2-cell bounds $\delta_{\text{trop}}(i, \tau; W)$ (Appendix L), and

(ii) a confidence-weighted advisory term $\delta^{\text{spec}} = \omega \odot \widehat{\delta}_{\text{spec}}$ (Definition M.7). All contributions enter the post-collapse window budget via the quantale sum \oplus (Definition M.11) and are measured in the fixed order “for each $t \rightarrow \mathbf{P}_i \rightarrow \mathbf{T}_\tau \rightarrow \text{compare}$.” With Summability (Appendix J), windowwise certificates paste across locally finite covers. Accordingly, the dictionary remains a reproducible, non-gating aid to certified after-collapse diagnostics, aligned with Appendices K/L/N and the global run protocol of Appendix G.

Appendix Q. p -adic Definable Windows (Denef–Pas) [Spec; finite event decomposition; finite Čech checks; local comparisons after collapse] (canon-aligned)

Standing conventions (canon). Let K be a non-archimedean local field with valuation $\text{val} : K^\times \rightarrow \mathbb{Z}$, residue field k , and angular component $\text{ac} : K \rightarrow k \cup \{0\}$. We work in the three-sorted Denef–Pas structure $(\text{VF}, \text{RF}, \text{VG}) = (K, k, \mathbb{Z})$ with Presburger arithmetic on VG . This appendix is [Spec]: it provides a *definable windowing interface* on valuation-time VG compatible with the global run protocol.

Global run protocol (non-negotiable). All numeric comparisons are evaluated *after collapse* in the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare on } \mathbf{T}_\tau \mathbf{P}_i},$$

with the same MECE, right-open windows and the same τ as B-Gate⁺ and the δ -ledger (Appendix G/J/N). All defects live in a fixed commutative quantale V with sum \oplus , order \preceq , and scalar action \odot (Appendix K/L/S), and are recorded in `run.yaml` (Appendix G).

Embedding VG into the global time axis. To align valuation-time with the global real parameter, fix once per run an affine embedding

$$\iota : \text{VG} = \mathbb{Z} \hookrightarrow \mathbb{R}, \quad \iota(n) := t_0 + s \cdot n, \quad s > 0,$$

and interpret all VG -windows on the real axis via ι . All window endpoints remain *right-open* under ι . (Changing (t_0, s) is a reindexing choice and must be logged.)

Q.1. Denef–Pas definable windows (right-open, MECE-ready)

Definition N.1 (DP-definable right-open window on VG). A *right-open window* $W \subset \text{VG}$ is a subset of \mathbb{Z} definable in Denef–Pas whose VG -projection admits a finite disjoint-union representation

$$W = \bigsqcup_{m=1}^M \left\{ n \in \mathbb{Z} \mid a_m \leq n < b_m, \ n \equiv r_m \pmod{c_m} \right\},$$

with $a_m \in \mathbb{Z} \cup \{-\infty\}$, $b_m \in \mathbb{Z} \cup \{+\infty\}$, $c_m \in \mathbb{Z}_{>0}$, $r_m \in \mathbb{Z}/c_m\mathbb{Z}$. Via ι , we regard W as a right-open subset of the global time axis.

Remark N.2 (Uniform families). For a DP-definable parameter set Λ , a family $\{W_\lambda\}_{\lambda \in \Lambda}$ is *uniformly* DP-definable if it admits a decomposition as in Definition N.1 with Presburger-definable data $a_m(\lambda)$, $b_m(\lambda)$, $r_m(\lambda)$, $c_m(\lambda)$. All finiteness bounds below then hold uniformly in λ as *statements about the VG-sort*.

Definition N.3 (Canonical MECE refinement on bounded valuation ranges). Fix a bounded valuation range $I = [A, B) \subset \text{VG}$ with $A, B \in \mathbb{Z}$. A *canonical MECE refinement* of I is a finite partition into right-open single-step windows

$$I = \bigsqcup_{n=A}^{B-1} [n, n+1),$$

transported to \mathbb{R} via ι . All audits and budgets on DP windows are ultimately reduced to this MECE refinement unless explicitly stated otherwise.

Q.2. Finite event decomposition and finite Čech checks (windowed)

Theorem N.4 ([Spec] DP windows on VG: finite events; finite overlap checks). *Let $W \subset VG$ be DP-definable and bounded, i.e. $W \subset I = [A, B)$ for some integers $A < B$. Then:*

1. **Finite event decomposition (after collapse).** *Any VG-indexed bookkeeping map used after collapse—including (i) bar endpoint readouts of $\mathbf{T}_\tau \mathbf{P}_i$, (ii) bin assignments, (iii) defect counters and ledger tags—is Presburger definable on VG. Hence on $W \subset [A, B)$ it has only finitely many value changes. Equivalently, on the canonical MECE refinement (Definition N.3) all such readouts are determined by finitely many cells/steps.*
2. **Finite Čech checks for the run cover.** *Under the global window policy (Appendix G/J), the cover used for audits on a bounded range is a finite family of right-open windows (typically the MECE refinement or a coarsening). Therefore the Čech nerve is finite and Overlap Glue terminates after finitely many checks (Appendix J).*
3. **Comparators are evaluated only after collapse.** *PF/BC/Control comparisons (Appendix N and any applicable control appendices) are evaluated only on $\mathbf{T}_\tau \mathbf{P}_i(F|_W)$ in the mandated order. Any residual slack (discretization/measurement or [Spec] commutation defects) is recorded as δ -terms and aggregated by \oplus in V .*
4. **Reproducibility.** *Event counts (or the MECE refinement size), overlap-check counts, and all window-wise budgets $\Sigma \delta_W(i)$ must be logged in run.yaml (Appendix G).*

Proof sketch. Presburger quantifier elimination yields piecewise-linear/constant behavior on VG; on a bounded subset of \mathbb{Z} this implies finitely many changes. The run protocol enforces a finite window family on bounded ranges; hence the nerve is finite and Overlap Glue performs finitely many comparisons. All comparisons are post-collapse by policy and use exactness/1-Lipschitz of \mathbf{T}_τ (Appendix A). \square

Remark N.5 (What “finite Čech” means here). We do *not* assume an arbitrary DP-definable cover has small nerve dimension. Instead, finiteness is ensured operationally by using a *finite run cover* on bounded ranges (Appendix G/J), typically derived from the MECE refinement (Definition N.3) and/or a finite coarsening recorded in run.yaml.

Q.3. V-nonexpansion and shift commutation (after collapse)

Proposition N.6 (Shift commutation and V-nonexpansion (post-collapse)). *Let S^ν be a V-Lawvere shift/reindexing operator (Chapter 2) acting on persistence by time translation/relabeling, and fix $\tau \geq 0$. Then on any window W (viewed on \mathbb{R} via ι) we have a canonical identification*

$$\mathbf{T}_\tau \circ S^\nu \cong S^\nu \circ \mathbf{T}_\tau,$$

and consequently, for any F, G ,

$$d_V(\mathbf{T}_\tau \mathbf{P}_i(F|_W), \mathbf{T}_\tau \mathbf{P}_i(G|_W)) \preceq d_V(\mathbf{P}_i(F|_W), \mathbf{P}_i(G|_W)).$$

All distances are measured after \mathbf{T}_τ ; any subsequent 1-Lipschitz post-processing on persistence cannot increase them (Appendix A/L).

Q.4. Local bridge trigger via E_1 on DP windows (after collapse)

Corollary N.7 (Local E_1 trigger on DP windows (approved direction)). *Let $W \subset \mathbf{VG}$ be a bounded DP-definable window, and fix $\tau \geq 0$. If the certified, post-collapse test $E_1(W) = 0$ holds for $\mathbf{T}_\tau \mathbf{P}_1(F|_W)$ (Appendix M/J), then*

$$E_1(W) = 0 \implies \mathbf{T}_\tau \mathbf{P}_1(F|_W) = 0 \implies \mathbf{PH}_1(\mathbf{T}_\tau F|_W) = 0 \implies \text{Ext}^1(\mathcal{R}(\mathbf{T}_\tau F)|_W, k) = 0,$$

where the last implication is the one-way bridge used only in its stated derived setting (Appendix C). No global converse and no additional equivalences are claimed.

Q.5. Quantitative commutation for DP transfers (Spec; after collapse)

Declaration N.8 (DP-transfer contract with 2-cell bound [Spec]). Let Trans be a DP-definable transfer/post-processing step (e.g. a p -adic tropicalization or functorial comparison) which is non-expansive up to a recorded defect on a window W . Fix $\tau \geq 0$. Record parameters $\kappa \in (0, 1]$ and $\delta_{\text{Tr}}(i, \tau; W) \in V$ such that, after \mathbf{T}_τ ,

$$d_V(\mathbf{T}_\tau \mathbf{P}_i(\text{Trans } F|_W), \mathbf{T}_\tau \mathbf{P}_i(\text{Trans } G|_W)) \preceq \kappa \odot d_V(\mathbf{T}_\tau \mathbf{P}_i(F|_W), \mathbf{T}_\tau \mathbf{P}_i(G|_W)) \oplus \delta_{\text{Tr}}(i, \tau; W).$$

Log κ and $\delta_{\text{Tr}}(i, \tau; W)$ in `run.yaml` and include them in the window budget $\Sigma \delta_W(i)$ (Appendix L/K/G).

Remark N.9 (No hidden commutation). No commutation beyond the recorded defect δ_{Tr} is assumed. If a stricter commutation statement is needed, it must be stated as a separate 2-cell bound and explicitly budgeted (Appendix L).

Q.6. P6 (Summability): $\bigoplus_W \Sigma \delta_W$ is finite on bounded DP ranges

Theorem N.10 (T-Delta-Sum-Converges (P6) on bounded DP ranges). *Let $I = [A, B) \subset \mathbf{VG}$ be bounded and let $\{W_j\}_{j \in J}$ be a locally finite family of DP-definable windows with $\bigcup_{j \in J} W_j \supset I$. Fix degree i and a pipeline whose per-window ledger yields $\Sigma \delta_{W_j}(i) \in V$ (Appendix G/L/K). Then the aggregated budget over I ,*

$$\bigoplus_{j \in J} \Sigma \delta_{W_j}(i),$$

is a finite \oplus -sum (hence convergent) in V .

Proof. Since $I = [A, B) \subset \mathbb{Z}$ is finite as a set of valuation levels, local finiteness implies that only finitely many windows W_j intersect I (otherwise some level $n \in I$ would lie in infinitely many W_j). Hence the index set $\{j \in J : W_j \cap I \neq \emptyset\}$ is finite, and the displayed \oplus -sum is finite. \square

Corollary N.11 (Unbounded ranges: explicit summability hypothesis [Spec]). *If $I = \mathbf{VG}$ is unbounded, P6 is enforced by an explicit summability hypothesis recorded in `run.yaml`, e.g.: there exist $\rho \in (0, 1)$ and $C \in V$ such that all per-level contributions satisfy $\delta_{\text{type}}(n) \preceq C \odot \rho^{|n|}$, or any other declared sufficient criterion ensuring $\bigoplus_n \delta(n)$ converges in V . Absent a logged summability rule, no global convergence claim is made.*

Remark N.12 (Uniformity across DP-families [Spec]). In DP-uniform families, any claimed parameter-uniform bounds (e.g. uniform C, ρ for Corollary N.11) must be stated as [Spec] and logged explicitly. No analytic uniformity (e.g. “uniform in p ”) is assumed unless recorded with an auditable δ -budget.

Q.7. Reproducibility (run.yaml) for DP mode

```

policy:
  after_collapse_only: true
definable:
  structure: "Denef-Pas"
  field: "Q_p"          # or finite extension; must specify
  uniformizer: "pi"
  vg_embedding:
    t0: 0.0
    scale: 1.0          # s>0; t = t0 + s*n
windows:
  mece: true
  right_open: true
  vg_windows:
    - "A <= n < B"          # bounded valuation range I
    - "a1 <= n < b1 and n % c1 == r1" # optional coarsenings
checks:
  finite_events: true
  finite_overlap: true
two_cell:
  dp_transfer:
    kappa: "<0<kappa<=1>"
    delta_Tr: "<rule or per-(i,tau,W) table>"
ledger:
  channels: ["disc", "meas", "Tr", "AB", "spec"]

```

Q.8. Minimal working recipe (Spec; non-gating)

1. **Fix DP windows.** Specify $I = [A, B) \subset VG$ and any DP-definable coarsenings $W \subset I$ (Definition N.1); log the formulas and ι .
2. **Reduce to MECE if needed.** Refine to the canonical MECE partition into $[n, n+1)$ (Definition N.3) whenever budgets or Overlap Glue require normalization.
3. **Compute post-collapse readouts.** For each window W and degree i , compute $\mathbf{T}_\tau \mathbf{P}_i(F|_W)$ in the mandated order; extract event counts (Theorem N.4).
4. **Run certified local tests.** If triggered, run $E_1(W) = 0$ after collapse; when verified, apply Corollary N.7 in the approved direction.
5. **Apply transfers only with budgets.** If a DP-transfer is used, record κ and $\delta_{Tr}(i, \tau; W)$ and include them in $\Sigma \delta_W(i)$ (Declaration N.8).
6. **P6 on bounded ranges.** On bounded I , convergence of the aggregated budget is automatic under local finiteness (Theorem N.10); on unbounded ranges, enforce and log an explicit summability rule (Corollary N.11).

Q.9. Non-claims (hard boundaries)

No pre-collapse control; no commutation beyond recorded 2-cell bounds; no equivalence beyond the one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ in $D^b(k\text{-mod})$. No analytic uniformity across primes/fields is assumed unless stated as [Spec] with explicit, audited δ -budgets.

Q.10. Integration points (scope-safe)

This appendix supplies a DP-definable VG-window interface compatible with: (i) the global MECE/right-open window policy and reproducible run protocol (Appendix G), (ii) finite Overlap Glue checks on bounded windows (Appendix J), (iii) post-collapse comparison discipline and budget accounting (Appendix L/K/N), and (iv) local bridge triggers via certified after-collapse tests (Appendix C/M). Any arithmetic/motivic/Langlands or other external uses remain [Spec] and must route through explicit 2-cell bounds and δ -ledger entries; nothing in this appendix upgrades such applications to certified results.

Appendix R. Iwasawa–AK Interface [Spec; Control \Rightarrow Overlap Gate & μ -alignment guidelines] (canon-aligned)

Standing conventions (canon). Fix a prime p . Let $\Gamma \simeq \mathbb{Z}_p$ be the Galois group of a cyclotomic (or other one-parameter) tower K_∞/K with finite layers K_n and open subgroups $\Gamma_n = \Gamma^{p^n}$. Write $\Lambda_{\text{Iw}} = \mathbb{Z}_p[[\Gamma]]$. Let $(M_n)_{n \geq 0}$ be a compatible system of finite p -primary modules (e.g. class groups or Selmer quotients), and $M_\infty = \varprojlim M_n$ a cofinitely generated Λ_{Iw} -module.

Mandatory measurement order (after-collapse). All persistence measurements and numeric comparisons are evaluated *after collapse* in the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare on } \mathbf{T}_\tau \mathbf{P}_i},$$

with the same MECE, right-open windows and the same τ as B-Gate⁺ and the δ -ledger (Chapter 1; Appendix G/J/N). Truncation \mathbf{T}_τ is exact and 1-Lipschitz (Appendix A). All defects aggregate in a fixed commutative quantale V with sum \oplus , order \preceq , and scalar action \odot (Appendix K/L/S), and are recorded in run.yaml (Appendix G).

Definable windows and finite checks. Windows are definable either o-minimally (real side) or in Denef–Pas (Appendix Q). All runtime covers used for gates/audits on bounded ranges are *finite* (or locally finite with explicit P6 logging), with finite Overlap Glue checks as per Appendix J/Q. Any statement in this appendix that depends on definability/coverage is [Spec] and must be backed by a logged cover specification and event counts.

Finite means finite p -group. Every reference to a kernel/cokernel being *finite* means a finite p -group.

R.1. Control maps and the arithmetic Overlap Gate (post-collapse)

Definition O.1 (Control map). A *control map* at level n is a natural comparison

$$\Phi_n : (M_\infty)_{\Gamma_n} \longrightarrow M_n,$$

with finite kernel and cokernel, where $(-)_{\Gamma_n}$ denotes coinvariants.

Definition O.2 (Control defect magnitude). Define the (dimensionless) control defect magnitude

$$e_{\text{ctrl}}(n) := v_p|\ker \Phi_n| + v_p|\text{coker } \Phi_n| \in \mathbb{R}_{\geq 0}.$$

Definition O.3 (Budget embedding). Fix a monotone embedding $\eta : \mathbb{R}_{\geq 0} \rightarrow V$ (e.g. $\eta(x) = x$ for $V = ([0, \infty], +, 0)$). The *control budget* recorded on a window W is

$$\delta_{\text{ctrl}}(n; W) := \eta(e_{\text{ctrl}}(n)) \in V.$$

By default $\delta_{\text{ctrl}}(n; W)$ is taken constant in W (control is arithmetic-level), but a window-local version is permitted if explicitly logged and justified by the pipeline interface used.

Definition O.4 (Arithmetic Overlap Gate (AK; post-collapse)). Let $\{W_\alpha\}$ be a *finite* definable cover of a bounded time window (Appendix G/J/Q), with overlap multiplicity

$$m := \sup_t \#\{\alpha : t \in W_\alpha\} \quad (\text{finite; logged in run.yaml}).$$

An *Arithmetic Overlap Gate* for a comparison $(F \Rightarrow G)$ requires that for every overlap $W_{\alpha\beta} := W_\alpha \cap W_\beta$, the post-collapse comparison map

$$\vartheta_{\alpha\beta} : \mathbf{T}_\tau \mathbf{P}_i(F|_{W_{\alpha\beta}}) \longrightarrow \mathbf{T}_\tau \mathbf{P}_i(G|_{W_{\alpha\beta}})$$

is an isomorphism in $\text{Pers}_k^{\text{ft}}$ up to a *finite p -group discrepancy* whose size is recorded as a ledger term $\delta^{\text{alg}}(i, \tau; W_{\alpha\beta}) \in V$. All such δ^{alg} aggregate by \oplus in the window budget.

Remark O.5 (Why we ledger finite discrepancies). Finite p -group kernels/cokernels are invisible to generic-dimension diagnostics (μ, u in Appendix D) but are *not* ignored operationally: they are recorded as δ^{alg} and charged to the window budget. This separates “Type IV” (invisible colimit defects) from finite arithmetic control errors: both can be invisible to (μ, u) , but both must be budgeted when they affect comparisons.

Proposition O.6 (Control \Rightarrow Arithmetic Overlap Gate [Spec]). *Assume:*

1. A control system $\{\Phi_n\}$ as in Definition O.1.
2. A definable finite cover $\{W_\alpha\}$ on the time axis with overlap multiplicity m (Definition O.4).
3. An [Spec] AK–Iwasawa interface that assigns to arithmetic objects a comparison pair $(F \Rightarrow G)$ in the pipeline such that, after the mandated order $t \rightarrow \mathbf{P}_i \rightarrow \mathbf{T}_\tau \rightarrow \text{compare}$, the induced overlap discrepancies are bounded by the control defect magnitude:

$$\delta^{\text{alg}}(i, \tau; W_{\alpha\beta}) \preceq \delta_{\text{ctrl}}(n; W_{\alpha\beta}) \quad \text{for all overlaps } W_{\alpha\beta}.$$

Then the Arithmetic Overlap Gate holds (Definition O.4), and moreover the total algebraic budget over the cover satisfies

$$\bigoplus_{\alpha\beta} \delta^{\text{alg}}(i, \tau; W_{\alpha\beta}) \preceq m \odot \sup_{\alpha\beta} \delta_{\text{ctrl}}(n; W_{\alpha\beta}).$$

Proof idea. All comparisons are post-collapse by policy; exactness of \mathbf{T}_τ preserves the finite-defect nature of arithmetic discrepancies. The bound $\delta^{\text{alg}} \preceq \delta_{\text{ctrl}}$ is the interface axiom. Since overlap multiplicity is m , each timepoint participates in at most m overlaps; quantale monotonicity and commutativity yield the stated aggregation bound. \square

Corollary O.7 (Finite glue with budget [Spec]). *Under Proposition O.6, Overlap Glue (Appendix J) performs finitely many overlap checks on the bounded run cover and yields a ledgered algebraic budget*

$$\delta_{\text{alg}}^{\text{tot}}(i, \tau) \preceq m \odot \sup_{\alpha\beta} \delta_{\text{ctrl}}(n; W_{\alpha\beta}),$$

which must be included in $\Sigma\delta_W(i)$ for the relevant windows.

Remark O.8 (Cofinal tails). Replacing (M_n) by a cofinal subsequence in n preserves the Overlap Gate verdict and the AK tower diagnostics (μ, u) computed after \mathbf{T}_τ (Appendix D/J), provided the interface and window policy are unchanged and the ledger is updated accordingly.

R.2. μ -alignment guidelines (window-local; non-identification by default)

Let μ_{class} denote the classical Iwasawa μ -invariant of the torsion Λ_{Iw} -module M_∞ (when defined). Let μ_{Collapse} denote the AK diagnostic μ computed *after* \mathbf{T}_τ on windows from the tower comparison map (Appendix D/J).

Declaration O.9 (μ -alignment decision tree (Spec; window-local; verification-first)). Work on a definable run cover with logged overlap multiplicity m and finite checks; all quantities below are post-collapse and budgets aggregate in V .

1. **Precheck (torsion).** If M_∞ has nonzero Λ_{Iw} -rank, declare $\mu_{\text{class}} = \text{NA}$ and do *not* attempt alignment; report the free rank separately.
2. **Control regime classification.** Compute/declare the control trend from $e_{\text{ctrl}}(n)$ (Definition O.2):
 - *Bounded control:* $\sup_n e_{\text{ctrl}}(n) < \infty$.
 - *Sublinear control:* $e_{\text{ctrl}}(n) = o(p^n)$.
 - *Uncontrolled:* otherwise.
3. **Pipeline regime classification.** Classify the post-collapse pipeline steps on the window as:
 - *Deletion-only:* only deletion-type steps (Appendix E/O) and 1-Lipschitz post-processing occur after collapse.
 - *General:* inclusion-type amplification may occur (no monotone claim; stability only).
4. **Alignment outcomes (guidelines; not automatic identifications).**
 - (a) *Coincidence (verified):* If bounded control *and* deletion-only pipeline, then alignment $\mu_{\text{Collapse}} \approx \mu_{\text{class}}$ is *permitted as a status* only after a cofinal-tail regression check succeeds on the run cover, with tolerance derived from the ledger (see Remark O.10).
 - (b) *Lower-bound mode:* Under bounded control alone, report μ_{Collapse} as a [Spec] *upper envelope* of persistent obstruction; do not claim equality unless verified. (Heuristic expectation: inclusion-type amplification can inflate μ_{Collapse} , hence $\mu_{\text{class}} \preceq \mu_{\text{Collapse}}$ is a plausible directional check, but remains [Spec] unless verified.)
 - (c) *Bracket mode:* Under sublinear control, define the drift rate

$$\mu_{\text{drift}} := \limsup_{n \rightarrow \infty} \frac{e_{\text{ctrl}}(n)}{p^n} \in \mathbb{R}_{\geq 0},$$

and report a window-local bracket statement only in the form:

$$“\mu_{\text{Collapse}} \text{ is consistent with } \mu_{\text{class}} \text{ up to drift } \mu_{\text{drift}}”,$$

together with the logged μ_{drift} and the observed δ_{ctrl} trend; no numeric inequality is promoted to certified status unless verified.

- (d) *No-claim mode:* Under uncontrolled growth of $e_{\text{ctrl}}(n)$, do not align; report μ_{Collapse} together with the ledger trend and the failure reason (`mu_decision.status = no_claim`).

All outcomes are **window-local** and are **non-identifications** unless explicitly marked *Coincidence (verified)*.

Remark O.10 (Verification hook for “Coincidence”). A minimal verification routine on a cofinal tail $\{n_j\}$ is: (i) compute an empirical growth proxy $\widehat{\mu}(n_j)$ from the chosen arithmetic observable (e.g. $p^{-n_j} v_p |M_{n_j}|$ if available and logged); (ii) compute μ_{Collapse} from the AK diagnostics on the same windows (after \mathbf{T}_τ); (iii) accept “Coincidence (verified)” only if the discrepancy stays below a tolerance derived from the full window budget $\Sigma\delta_W(i)$ and the control budget δ_{ctrl} , for all windows in the run cover. The exact observable and tolerance rule must be declared in `run.yaml`.

Remark O.11 (What each μ measures). μ_{class} governs p -power growth in the classical Iwasawa asymptotic (when applicable). μ_{Collapse} is a post-collapse obstruction diagnostic derived from the AK tower comparison map after truncation (Appendix D/J). This appendix provides *alignment guidelines* and logging requirements; it does not upgrade any arithmetic identification to a certified theorem without an explicit verification routine and budget justification.

R.3. Gate coupling and Restart/Summability (P6)

Definition O.12 (Gate coupling (Overlap Gate + B-Gate⁺)). Couple the Arithmetic Overlap Gate with B-Gate⁺ (Appendix J). On a window W and degree i , B-Gate⁺ passes if

$$\text{gap}_\tau(i) > \Sigma\delta_W(i),$$

where $\Sigma\delta_W(i)$ includes $\delta_{\text{ctrl}}(n; W)$, all overlap-glue algebraic terms δ^{alg} , and any other pipeline terms (Mirror/Transfer, A/B residuals, discretization/measurement).

Proposition O.13 (Restart and P6 (window pasting)). *Assume consecutive windows are linked only by deletion-type steps and ε -continuations measured post-collapse, and the run satisfies P6: $\sum_W \Sigma\delta_W(i) < \infty$ (Appendix J; bounded DP ranges satisfy finiteness as per Appendix Q). Then the Restart inequality of Appendix J propagates a positive safety margin across windows, and windowwise certificates paste to a global certificate on $\bigcup W$.*

R.4. Reproducibility hooks (run.yaml)

```
policy:
  after_collapse_only: true
iwasawa:
  prime: p
  tower: "cyclotomic"          # description
  gamma_presentation: "Z_p"    # generator choice logged
  levels: {start: n0, end: n1}
  control:
    e_ctrl_rule: "vp(|ker Phi_n|)+vp(|coker Phi_n|)"
    bounded: true/false
    sublinear: true/false
    eta: "<monotone embedding R>=0 -> V>"
    delta_ctrl: "<rule or table>"
  overlap_gate:
    cover: "<o-minimal | Denef-Pas>"
    windows: "<MECE, right-open spec>"
    overlap_multiplicity_m: "<integer>"
    finite_checks: true
  ledger:
```

```

channels: ["ctrl", "alg", "disc", "meas", "Tr", "AB", "spec"]
b_gate_plus:
  tau: "<value>"
  gap_tau: "<per window/degree>"
  passed: true/false
mu_alignment:
  classical_mu: "<value | NA>"
  collapse_mu: "<value | per-window table>"
  pipeline_regime: "deletion_only | general"
  status: "NA | coincidence_verified | lower_bound | bracket | no_claim"
  drift_mu: "<value if bracket>"
  verification:
    observable: " $p^{\{-n\}}$  vp(|M_n|)" # or declared alternative
    tolerance_rule: "<uses Sigma_delta and delta_ctrl>"
tests:
  T_Iwasawa: true

```

R.5. Minimal decision routine (Spec; non-gating)

```

# inputs: classical_mu (or NA), e_ctrl(n), pipeline_regime
def mu_alignment(classical_mu, e_ctrl, pipeline_regime, verified):
  if classical_mu == "NA":
    return ("NA", None)
  bounded = sup_n(e_ctrl[n]) < ∞
  sublinear = limsup_n(e_ctrl[n] / p**n) == 0
  if bounded and pipeline_regime == "deletion_only" and verified:
    return ("coincidence_verified", 0)
  if bounded:
    return ("lower_bound", 0) # guideline only unless verified
  if sublinear:
    drift = limsup_n(e_ctrl[n] / p**n)
    return ("bracket", drift) # consistency statement only
  return ("no_claim", None)

```

R.6. Edge cases and guard-rails

- **Non-torsion modules.** If $\text{rank}_{\Lambda_{\text{Iw}}} M_\infty > 0$, set $\mu_{\text{class}} = \text{NA}$ and do not align.
- **Inclusion-type steps.** Any inclusion-type amplification (post-collapse) forbids automatic “Coincidence”; require verification or fall back to Lower-bound/Bracket/No-claim.
- **Overlap multiplicity.** The bound m must be computed from the *actual run cover* and logged; do not assume $m \leq 2$ unless the cover is explicitly MECE/right-open with the corresponding overlap structure.
- **Character twists.** Alignment routines may be repeated under finite character twists; twists must be logged as part of the arithmetic configuration.
- **Cofinal tails.** Replacing (M_n) by a cofinal tail preserves window policies and diagnostics provided the ledger is recomputed and the same τ and windows are used.

R.7. Formalization stubs (Lean/Coq) [Spec]

```

structure ControlMap (n : ℕ) :=
  (phi : coinv Γ_n M∞ → M_n)
  (ker_fin : finite_p (kernel phi))
  (coker_fin : finite_p (cokernel phi))

def e_ctrl (φ : ControlMap n) : ℝ≥0 :=
  vp (kernel φ).card + vp (cokernel φ).card

def δ_ctrl (η : ℝ≥0 → V) (φ : ControlMap n) : V :=
  η (e_ctrl φ)

-- Interface axiom: post-collapse algebraic defects are bounded by δ_ctrl
axiom iwasawa_AK_interface :
  ∀ (φ : ControlMap n) (W : Window) (i : ℤ) (τ : ℝ≥0),
    δ_alg(i,τ,W) ≤ δ_ctrl η φ

theorem control_implies_overlap_gate :
  ... -- derives Proposition R:prop:control-overlap

```

R.8. Summary

Control maps with finite kernel/cokernel yield, via an explicit [Spec] AK–Iwasawa interface, an Arithmetic Overlap Gate after collapse. Finite discrepancies are budgeted as δ^{alg} using a monotone embedding η of p -exponents into the quantale V , and aggregated over a finite definable run cover with logged overlap multiplicity m . Gate coupling with B-Gate⁺ is performed windowwise using the unified budget $\Sigma\delta_W(i)$ (including δ_{ctrl}), and global certificates paste under Restart/Summability (Appendix J; Appendix Q for DP ranges). Finally, μ -alignment is treated as a *window-local guideline*: identification beyond “Coincidence (verified)” is not asserted without an explicit regression check and ledger-based tolerance rule recorded in `run.yaml`.

Appendix S. Quantale Catalog [Spec; selection policy for the δ -ledger & V -metrics] (canon-aligned)

Standing conventions (canon). All *quantitative* measurements are evaluated *after collapse*, i.e. in the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare on } \mathbf{T}_\tau \mathbf{P}_i},$$

using the same MECE, right-open windows and the same τ as elsewhere (Chapter 1; Appendix G/J/N). Truncation \mathbf{T}_τ is exact and 1-Lipschitz for the interleaving metric d_{int} (Appendix A). All defects are recorded in the δ -ledger (Appendix G) as V -valued entries (channels such as δ^{alg} , δ^{disc} , δ^{meas} , δ^{spec} , and typed subchannels δ^{Tr} , δ^{Fun} , Δ_{comm} as used in Appendices L/M/P/Q/R). Unless explicitly declared otherwise (see §P), a *single base quantale* V is fixed for the run and used for all aggregation by \oplus . Finite definable covers on bounded windows (Appendix H/J/Q) ensure that all joins and ledger aggregations used for runtime gating are *finite*; countable covers are permitted only under P6/Summability with explicit logging (Appendix J/Q).

S.1. Implementable-range axioms (finite commutative quantales)

A (*finite*) *commutative quantale* for this pipeline is a tuple $(V, \oplus, 0, \preceq, \vee)$ such that:

(Q0) (V, \preceq) is a poset admitting *finite joins* \vee (we do *not* assume completeness).

(Q1) $(V, \oplus, 0)$ is a commutative monoid; $a \oplus 0 = a$.

(Q2) Monotonicity: $a \preceq a', b \preceq b' \Rightarrow a \oplus b \preceq a' \oplus b'$.

(Q3) Finite distributivity: $a \oplus (b \vee c) = (a \oplus b) \vee (a \oplus c)$.

(Q4) (Optional) scalar action \odot by a commutative monoid R (typically $R = [0, 1]$ or $\mathbb{R}_{\geq 0}$) such that $r \odot (a \vee b) = (r \odot a) \vee (r \odot b)$ and $r \odot (a \oplus b) = (r \odot a) \oplus (r \odot b)$. This is used for contraction factors $\kappa \in (0, 1]$ and confidence weights $\omega \in [0, 1]$ (Appendices L/M/P/Q).

Definition P.1 (*V*-Lawvere distance). A *Lawvere V-distance* on a set X is a map $d_V : X \times X \rightarrow V$ with $0 \preceq d_V(x, x)$ and the triangle inequality

$$d_V(x, z) \preceq d_V(x, y) \oplus d_V(y, z).$$

A map $f : (X, d_V) \rightarrow (Y, d'_V)$ is *V-non-expansive* if $d'_V(fx, fy) \preceq d_V(x, y)$ for all x, y .

Remark P.2 (From real metrics to V). In many modules (e.g. persistence comparisons), a real-valued bound $d_{\text{int}} \leq \varepsilon$ is converted to a V -bound by a declared monotone embedding $\eta : \mathbb{R}_{\geq 0} \rightarrow V$ (Remark P.3). This keeps the *after-collapse* 1-Lipschitz claims of \mathbf{T}_τ valid after change of codomain: if $d_V := \eta \circ d_{\text{int}}$, then \mathbf{T}_τ remains V -non-expansive (Appendix A).

S.2. Catalog of canonical base quantales (all [Spec])

S.2.1. Additive budget (V_{add}).

$$V_{\text{add}} := ([0, \infty], \oplus = +, 0, \preceq = \leq, \vee = \max), \quad R = \mathbb{R}_{\geq 0}, \quad r \odot a := r \cdot a.$$

Intended semantics: ledger entries are radii/budgets that accumulate; \oplus is literal addition. *Use when*: many small contributors must be accounted for (continuation radii, discretization/measurement tolerances, algebraic totals, Restart/Summability budgets; Appendix J).

S.2.2. Worst-case budget (V_{max}).

$$V_{\text{max}} := ([0, \infty], \oplus = \max, 0, \preceq = \leq, \vee = \max), \quad R = [0, 1], \quad r \odot a := r \cdot a.$$

Intended semantics: acceptance is controlled by the largest spike; aggregation retains only the worst defect. *Use when*: a single dominant defect controls pass/fail (tightest overlap, maximal commutation defect, worst bin; sup-style audits).

S.2.3. Magnitude \times confidence (V_{prob}). Let $V_{\text{prob}} := [0, \infty] \times [0, 1]$ with product order $(x, c) \preceq (x', c') \iff x \leq x' \wedge c \leq c'$, finite join \vee given coordinatewise by (\max, \max) , and monoid law

$$(x, c) \oplus (y, d) := (x + y, c \cdot d), \quad 0 := (0, 1).$$

Optionally take scalar action $r \odot (x, c) := (rx, c)$ for $r \in [0, 1]$. *Intended semantics*: each defect carries a magnitude and an independent confidence/success weight; magnitudes add, confidences multiply. *Use when*: advisory or estimator outputs must be tracked explicitly (e.g. Appendix P confidence ω , laboratory-style regression channels). *Guard-rail*: this is [Spec] and should be used only when the independence/multiplicativity convention is declared in run.yaml.

S.2.4. Product quantales (multi-channel bases). Given base quantales V_1, V_2 , define

$$V_1 \times V_2, \quad (a_1, a_2) \oplus (b_1, b_2) := (a_1 \oplus_1 b_1, a_2 \oplus_2 b_2),$$

ordered coordinatewise; finite joins are coordinatewise. *Use when:* two audit channels must be retained without collapsing information, e.g. $V_{\text{add}} \times V_{\text{max}}$ (cumulative amount *and* worst spike), or $V_{\text{add}} \times V_{\text{prob}}$ (budget with explicit confidence).

Remark P.3 (Embedding of numeric budgets). Arithmetic/control numbers (e.g. $v_p | \ker | + v_p | \text{coker} |$ in Appendix R) and real-valued radii (e.g. ε in continuation bounds) are injected by a fixed monotone $\eta : \mathbb{R}_{\geq 0} \rightarrow V$:

$$\eta(x) = x \text{ for } V_{\text{add}}, V_{\text{max}}; \quad \eta(x) = (x, 1) \text{ for } V_{\text{prob}},$$

and componentwise for products. The choice of η is part of the *base quantale declaration* in `run.yaml`.

S.3. Interoperability with the pipeline (after-collapse only)

- **Ledger aggregation.** The δ -ledger stores V -valued entries per window/degree and aggregates by \oplus to produce $\Sigma \delta_W(i)$ (Appendix G/J). Typed entries from Mirror/Transfer/Tropical 2-cells are logged as δ^{Tr} , δ^{Fun} , or δ^{trop} and contribute to $\Sigma \delta$ (Appendices L/M/P/Q).
- **V-metrics and shifts.** A V -distance d_V induces V -shifts S^v as in Chapter 2; tests $T_{V\text{shift}}$ enforce post-collapse commutation $\mathbf{T}_\tau \circ S^v \simeq S^v \circ \mathbf{T}_\tau$ and V -nonexpansion on $\mathbf{T}_\tau \mathbf{P}_i$.
- **Towers and diagnostics.** Tower comparisons and diagnostics (μ, u) are computed *after* \mathbf{T}_τ (Appendix D/J). The quantale governs how residuals accumulate in Overlap Glue and Restart/Summability budgets (Appendix J).
- **Arithmetic layers.** Finite kernel/cokernel control errors are embedded into V via η and charged as δ_{ctrl} or δ^{alg} depending on the interface (Appendix R). PF/BC residuals are entered as δ^{disc} , δ^{meas} (Appendix N).

S.4. Selection policy (run-level base; window-local objectives)

Policy. Choose a *base quantale* V *per run* (not ad hoc per window), and record it in `run.yaml.quantale`. Window-local objectives (safety vs. accounting vs. confidence) should be handled either by (i) selecting an appropriate *product base* (e.g. $V_{\text{add}} \times V_{\text{max}}$), or (ii) applying a safe change-of-base morphism (see §P) and logging the coercion.

Guidelines (objective \Rightarrow recommended base).

1. **Safety-margin dominated** (“*did anything spike?*”): use V_{max} or include a max-channel via $V_{\text{add}} \times V_{\text{max}}$.
2. **Cumulative risk accounting** (“*how much budget remains?*”): use V_{add} .
3. **Confidence bookkeeping** (advisory/estimator outputs): use V_{prob} or $V_{\text{add}} \times V_{\text{prob}}$.
4. **Dual-view audits** (amount *and* worst spike): use $V_{\text{add}} \times V_{\text{max}}$.

S.5. Change of base (safe coercions)

A *quantale morphism* $\phi : (V, \oplus, 0, \preceq) \rightarrow (W, \boxplus, 0', \preceq')$ is a monotone map satisfying

$$\phi(0) = 0', \quad \phi(a \oplus b) \preceq' \phi(a) \boxplus \phi(b).$$

Such ϕ is a *safe coercion*: it can only weaken (never strengthen) budget inequalities.

Example P.4 (Standard safe coercions).

$$V_{\text{add}} \rightarrow V_{\text{max}} : \phi(x) = x \quad (\text{since } x + y \geq \max\{x, y\}), \quad V_{\text{prob}} \rightarrow V_{\text{add}} : \phi(x, c) = x.$$

For product tracking:

$$V \rightarrow V \times W : a \mapsto (a, \phi'(a)),$$

where ϕ' is any quantale morphism into W .

Proposition P.5 (Safe coercion monotonicity). *If ϕ is a quantale morphism and $\Sigma\delta \in V$ is a ledger total, then for any gate threshold $\text{gap} \in V$,*

$$\Sigma\delta \preceq \text{gap} \implies \phi(\Sigma\delta) \preceq' \phi(\text{gap})$$

in W .

Remark P.6 (How to use change-of-base operationally). If a run fixes V_{add} but a report requires worst-case semantics, apply $\phi : V_{\text{add}} \rightarrow V_{\text{max}}$ to the *already computed* ledger totals and thresholds. Do *not* recompute budgets under a different aggregation rule without declaring a new base quantale and rerunning the ledger. All coercions used for reporting must be listed under `quantale.change_of_base`.

S.6. Reproducibility keys (mandatory)

quantale:

```

name: "V_add"           # V_add | V_max | V_prob | V_addxV_max | ...
op: "+"                 # "+", "max", "(+, ·)", "product"
unit: "0"               # "0" or "(0,1)" etc.
order: "<="              # coordinatewise when product
join: "max"              # finite join; coordinatewise if product
scalar_action:
  enabled: true
  monoid: "[0,1]"        # or "R>=0"
  rule: "r ⊙ a"           # declared action; per-base
eta_embed: "x -> x"     # or "x -> (x,1)" etc.
change_of_base:
  - {to: "V_max", phi: "x -> x"} # example safe coercion

```

tests:

```

T_Vshift: true
T_Vsubadd: true          # ledger aggregation law checks
T_ChangeOfBase: true     # verifies listed φ are morphisms

```

S.7. Non-claims (scope limits)

We do not assume completeness nor infinite distributivity beyond finite joins (definable finiteness suffices on bounded windows). No pre-collapse metric statements are made: all V -nonexpansion and V -Lipschitz claims are asserted only *after* \mathbf{T}_τ .

S.8. Integration map (where V is consumed)

Chapter 1 fixes the after-collapse evaluation order and ledger discipline. Chapter 2 defines V -metrics and V -shifts and supplies $T_{V\text{shift}}$. Appendix D/J compute tower diagnostics and pasting under Restart/Summability in the chosen V . Appendices L/M/P/Q/R inject 2-cell and arithmetic budgets via η and (optional) scalar actions \odot . Appendix N supplies PF/BC residual channels $\delta^{\text{disc}}, \delta^{\text{meas}}$. Appendix G/Chapter 12 enforce `run.yaml` compliance and test execution.

Appendix T. Implementation Notes / Notebooks [Spec; script skeletons, Gate Cascade, Convergence Manager, counterexample hunter, & CI demos]

Standing conventions (canon, enforced). All quantitative evaluations are *after collapse* in the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_t \implies \mathbf{T}_\tau \implies \text{compare on } \mathbf{T}_\tau \mathbf{P}_t},$$

with *one* declared window partition (MECE, right-open; Appendix G/H/J/Q) and a *single* τ per comparison bundle. All persistence modules are treated in the constructible range on bounded windows (Appendix A/H); filtered (co)limits are computed objectwise under the scope policy and returned to the constructible range when stated. \mathbf{T}_τ is exact, idempotent, and 1-Lipschitz (Appendix A). Deletion-type operations are the *only* operations granted monotone (nonincreasing) indicator claims; inclusion-type operations are *never* used for monotonicity, only for stability/nonexpansion when certified (Appendix E). All defects/budgets are written to a fixed commutative quantale V (Appendix S) via a δ -ledger with canonical channels

$$\delta^{\text{alg}}, \quad \delta^{\text{disc}}, \quad \delta^{\text{meas}}, \quad \delta^{\text{spec}},$$

and optional layer tags (e.g. $\delta^{\text{Tr}}, \delta^{\text{Fun}}, \delta^{\text{Gal}}$) that *refine* δ^{alg} (Appendix L/Q/R/S). Arithmetic overlaps use Control \Rightarrow Overlap Gate, with finite kernel/cokernel parts embedded into V via a fixed monotone $\eta : \mathbb{R}_{\geq 0} \rightarrow V$ (Appendix R/S). Notebook templates below are **[Spec]** scaffolds intended to *enable* Chapter 12 tests; they do not introduce new mathematical claims. Gate Cascade uses only after-collapse objects; the Convergence Manager enforces Restart/Summability (Appendix J) windowwise.

Remark Q.1 (Key normalization (mandatory)). To avoid drift between LaTeX notation and `run.yaml`, we use:

$$\text{delta}\hat{\text{alg}}, \text{delta}\hat{\text{disc}}, \text{delta}\hat{\text{meas}}, \text{delta}\hat{\text{spec}}$$

as canonical keys (Appendix G). Layer tags use `deltaTr`, `deltaFun`, `deltaGal`, etc., and must be aggregated into `deltaalg` or a declared `delta_total` according to the chosen quantale V (Appendix S).

T.1. Directory layout (minimal, reproducible)

```
akhdpst/
run.yaml          # manifest (Appendix G; single source of truth)
data/             # raw inputs (tropical/LMHS/p-adic/Selmer/etc.)
cache/           # intermediate artifacts (per window, per  $\tau$ , per i)
logs/            #  $\delta$ -ledger snapshots, audit trails, test outputs
notebooks/
  01_stage_tropical.ipynb
  02_stage_lmhs.ipynb
  03_stage_padic.ipynb
  04_stage_arithmetic.ipynb
```

```

05_convergence_manager.ipynb
06_gate_cascade.ipynb
07_counterexample_hunter.ipynb
demo_gl1_min.ipynb
demo_gl2_min.ipynb
scripts/
  tau_sweep.py
  mece_check.py
  ledger_aggregate.py
  convergence_manager.py
  gate_cascade.py
  hunter_generate.py
  demo_gl1.py
  demo_gl2.py
tests/
  test_vshift.py
  test_definable.py
  test_control_overlap.py
  test_gate_cascade.py
  test_convergence_manager.py
  test_hunter_regressions.py

```

T.2. Manifest fragments (per stage; consistent windows/ τ/i)

Record mandatory keys (Appendix G) with stage-local fields. *Window identifiers must match across all stages* (Appendix G/J), and all stage outputs must be indexed by (W, i, τ) .

T.2.1. Global: quantale, definability, windows, τ -sweep

```

quantale:
  name: "V_addxV_max"          # Appendix S
  op: "product"
  unit: "(0,0)"
  order: "coordinatewise"
  scalar_action: true
  eta_embed: "x->(x,x)"       # example; must match Appendix S choice
definable:
  structure: "R_an,exp"        # or "Denef-Pas" (Appendix Q)
  window_formulae:
    - "W1: a1 <= t < b1"
    - "W2: b1 <= t < b2"
windows:
  base: ["W1", "W2"]          # MECE target; enforced by tests
degrees:
  i_list: [0,1,2]             # explicit degrees audited
tau:
  grid: {start: 0.0, stop: 3.0, step: 0.1}
tests:

```

```

T_Vshift: true
T_Definable: true
T_Vsubadd: true

```

T.2.2. Mirror/Tropical \rightarrow advisory LMHS (Appendix P/L)

```

tropical:
  contraction_kappa: 0.9
  bins: [0, 0.25, 0.5, 1.0]
  export: "cache/tropical_{W}_{i}_{tau}.json"
lmhs:
  proxies: ["rankN", "weights", "h_infty"]
  omega_policy: "heldout|band|zero_default" # Appendix P
  export: "cache/lmhs_{W}_{i}_{tau}.json"
two_cell:
  bound_delta_trop_rule: "delta^Tr <= 0.05 per (W,i,tau)" # Appendix P/L

```

T.2.3. p-adic (Denef-Pas) transfer and arithmetic control (Appendix Q/R)

```

padic:
  structure: "Denef-Pas"
  field: "Q_p"
  p: 3
  uniformizer: "p"
  vg_scale: 1
  export: "cache/padic_{W}_{i}_{tau}.json"
iwasawa:
  tower: "cyclotomic"
  levels: {start: 0, end: 8}
  control_bounds:
    ker_vp_sup: 2
    coker_vp_sup: 3
  export: "cache/control_{W}_{i}_{tau}.json"
layered_delta:
  delta^Gal_rule: "eta(vpKer+vpCoker)" # Appendix R/S

```

T.2.4. Convergence Manager (Restart/Summability; Appendix J)

```

convergence:
  restart_policy:
    kappa: 1.0
    max_restarts_per_window: 2
  summability:
    tol_add_base: 1e-6 # tolerance in a declared V_add base (Appendix S)
    max_iters: 20
  stable_band_scan:
    neighborhood: 0.02
    min_band_width: 0.05

```

T.2.5. Gate Cascade (after-collapse only; Appendix J/N/R)

```
gate_cascade:
  order: ["B_GatePlus", "PF_BC", "OverlapGate"]
  B_GatePlus:
    gap_key: "gap_tau"          # measured after T_tau on the same (W,i,tau)
  PF_BC:
    enforce: true
    budget_keys: ["delta^disc", "delta^meas"]
  OverlapGate:
    enforce: true              # Appendix R: control overlap
    budget_key: "delta^Gal"
```

T.2.6. Counterexample hunter (regression generators; Appendix D/E/L/Q)

```
hunter:
  generators:
    - "typeIV_accumulation"    # near- $\tau$  pileup (Appendix D)
    - "mirror_comm_defect"     # large 2-cell  $\delta^{\text{Tr}}$  (Appendix L/P)
    - "inclusion_spike"        # inclusion-type stress (Appendix E)
  budget_caps:
    delta^Tr: 0.2
    delta^Fun: 0.2
    delta^Gal: 5
  export: "cache/hunter_{W}_{i}_{tau}.json"
```

T.2.7. CI demos (GL(1)/GL(2) minimal workflows; non-gating)

```
demo:
  gl1:
    tower: "cyclotomic"
    levels: {start: 0, end: 5}
    class_module: "toy_surrogate"    # explicitly non-analytic placeholder
  gl2:
    curve: "E:  $y^2 = x^3 - x$ "
    levels: {start: 0, end: 4}
    selmer_p: 3
```

T.3. τ -sweep driver (skeleton; canon order enforced)

```
# scripts/tau_sweep.py (pseudocode; after-collapse policy enforced)
from akhdpst import pipeline
```

```
cfg = pipeline.load_manifest("run.yaml")
```

```
for W in cfg.windows.base:
  cells = pipeline.definable_cells(W, cfg)          # finite by App. H/J/Q
  for i in cfg.degrees.i_list:
    for tau in pipeline.grid(cfg.tau.grid):
```

```

# (0) Initialize per-(W,i,tau) ledger frame
pipeline.ledger.begin_frame(W=W, i=i, tau=tau)

# (1) Mirror/Tropical stage: record only 2-cell / commutation budgets
trop = pipeline.tropical_readout(W, i, tau, cfg)
pipeline.ledger.add("delta^Tr", W, i, tau, trop.delta_Tr) # in V

# (2) LMHS proxies: advisory only (never a gate)
lmhs = pipeline.lmhs_proxy(trop, cfg)
# If a proposal yields a nonzero advisory term, store as delta^spec
pipeline.ledger.add("delta^spec", W, i, tau, lmhs.delta_spec)
pipeline.cache.save(lmhs, f"cache/lmhs_{W}_{i}_{tau}.json")

# (3) p-adic transfer / functorial comparison (Spec): record 2-cell budget
pad = pipeline.padic_readout(W, i, tau, cfg)
pipeline.ledger.add("delta^Fun", W, i, tau, pad.delta_Fun)

# (4) Canon object: compute persistence, then collapse, then compare
#   PiT means T_tau(P_i(F|_W)) in Pers^{ft}.
PiT = pipeline.read_Pi_then_Ttau(W, i, tau, cfg)

# (5) Certified tests triggered by advisory hints (Appendix P/Q):
E1 = pipeline.energy_bins_after_collapse(PiT, cfg)
if E1.is_zero():
    pipeline.bridge_certify_after_collapse(PiT, W, i, tau, cfg) # PH1=>Ext1

# (6) Arithmetic control / overlap budgets (Appendix R):
ctl = pipeline.control_bounds(W, i, tau, cfg)
pipeline.ledger.add("delta^Gal", W, i, tau, ctl.delta_Gal)

pipeline.ledger.end_frame(W=W, i=i, tau=tau)

pipeline.ledger.flush("logs/ledger.json")

```

T.4. Convergence Manager (Restart & stability bands; Appendix J)

All convergence decisions are made *after collapse* and must be compatible with Restart/Summability (Appendix J). Because a general quantale may not carry a numeric norm, tolerances are evaluated in a declared additive base via a logged change-of-base morphism (Appendix S).

```

# scripts/convergence_manager.py (pseudocode; Appendix J)
from akhdpst import pipeline, quantale

cfg = pipeline.load_manifest("run.yaml")

κ = cfg.convergence.restart_policy.kappa
tol = cfg.convergence.summability.tol_add_base

```

```

imax = cfg.convergence.summability.max_iters

# Coercion to an additive base for tolerance checks (logged in run.yaml)
phi_to_add = quantale.change_of_base(cfg.quantale, target="V_add")

def iterate_until_stable(W, i, tau0):
    tau = tau0
    prev = None

    for it in range(imax):
        PiT = pipeline.read_Pi_then_Ttau(W, i, tau, cfg)
        gap = pipeline.gap_tau(PiT, i, tau, cfg)          # B-Gate+ + margin
        Σδ = pipeline.ledger.window_total(W, i, tau, cfg) # ⊕-sum in V

        if quantale.lt(Σδ, gap, cfg.quantale):           # strict pass
            band = pipeline.find_stability_band(W, i, tau, cfg.convergence.stable_band_scan, cfg)
            return {"status": "passed", "W": W, "i": i, "tau": tau, "band": band,
                    "delta_total": Σδ, "gap": gap}

    # Restart update (Appendix J): adjust tau within a declared policy
    tau = pipeline.restart_update(tau, κ, cfg)

    # Tolerance check in additive base (Spec; operational stop)
    if prev is not None:
        a_now = phi_to_add(Σδ)
        a_prev = phi_to_add(prev)
        if abs(a_now - a_prev) <= tol:
            break
    prev = Σδ

    return {"status": "max_iters", "W": W, "i": i, "tau": tau, "delta_total": prev}

for W in cfg.windows.base:
    for i in cfg.degrees.i_list:
        info = iterate_until_stable(W, i, tau0=cfg.tau.grid.start)
        pipeline.cache.save(info, f"cache/convergence_{W}_{i}.json")

```

T.5. Gate Cascade (B-Gate⁺ → PF/BC → Overlap; after-collapse only)

The Gate Cascade is a *decision pipeline* on a fixed (W, i, τ) . All checks read only $T_\tau P_i$ objects and consult only ledger totals in V (Appendix J/N/R/S).

```

# scripts/gate_cascade.py (pseudocode; Appendix J/N/R; after-collapse)
from akhdpst import pipeline, quantale

def run_cascade(W, i, tau, cfg):
    PiT = pipeline.read_Pi_then_Ttau(W, i, tau, cfg)
    gap = pipeline.gap_tau(PiT, i, tau, cfg)          # after-collapse

```

```

 $\Sigma\delta$  = pipeline.ledger.window_total(W, i, tau, cfg)

# 1) B-Gate+ (Appendix J): accept only if gap >  $\Sigma\delta$ 
if not quantale.lt( $\Sigma\delta$ , gap, cfg.quantale):
    return {"stage": "B_GatePlus", "pass": False, "W": W, "i": i, "tau": tau,
            "gap": gap, "delta_total":  $\Sigma\delta$ }

# 2) PF/BC after-collapse comparators (Appendix N)
pfbc_ok, drift = pipeline.pfbc_check_after_collapse(PiT, W, i, tau, cfg)
pipeline.ledger.add("deltadisc", W, i, tau, drift.delta_disc)
pipeline.ledger.add("deltameas", W, i, tau, drift.delta_meas)
if not pfbc_ok:
    return {"stage": "PF_BC", "pass": False, "drift": drift}

# 3) Overlap Gate after-collapse (Appendix R): control overlap
ov_ok, ctl = pipeline.overlap_gate_after_collapse(W, i, tau, cfg)
pipeline.ledger.add("deltaGal", W, i, tau, ctl.delta_Gal)
return {"stage": "OverlapGate", "pass": ov_ok, "control": ctl, "W": W, "i": i, "tau": tau}

# notebook cell (06_gate_cascade.ipynb):
# res = run_cascade("W1", 1, tau, cfg); save_json(res, ...)

```

T.6. Counterexample hunter (adversarial generators; regression-only)

The hunter is a *stress testing* tool: it generates bounded-window instances designed to trigger known failure modes (Appendix D/E/L/Q). It never produces certificates; it only produces failing examples and associated ledger traces (Appendix G/J).

```

# scripts/hunter_generate.py (pseudocode; App. D/E/L/Q)
from akhdpst import pipeline

def gen_typeIV_accumulation(W, i, tau, density=50):
    # near- $\tau$  bar-length pileup (Type IV pattern), finite on bounded windows
    return pipeline.synthetic.typeIV_pileup(W, i, tau, density=density)

def gen_mirror_comm_defect(W, i, tau, target=0.15):
    # large Mirror Collapse (or Mirror Ctau) 2-cell defect (Appendix L/P)
    return pipeline.synthetic.mirror_defect(W, i, tau, target_delta=target)

def gen_inclusion_spike(W, i, tau, factor=2.0):
    # inclusion-type amplification: should break deletion-only monotonicity guards
    return pipeline.synthetic.inclusion_spike(W, i, tau, factor=factor)

def hunt_all(W, i, tau, cfg):
    cases = []
    for name, gen in [("typeIV", gen_typeIV_accumulation),
                     ("mirror", gen_mirror_comm_defect),
                     ("incl", gen_inclusion_spike)]:

```

```

F = gen(W, i, tau)
PiT = pipeline.to_after_collapse(F, W, i, tau, cfg)    # Pi then T_tau
report = pipeline.run_full_audit(PiT, W, i, tau, cfg)  # gate + ledger trace
cases.append({"name":name, "report":report})
pipeline.cache.save(report, f"cache/hunter_{name}_{W}_{i}_{tau}.json")
return cases

```

T.7. MECE window coverage check (test hook; Appendix G/H/J/Q)

```

# scripts/mece_check.py (pseudocode)
from akhdpst import windows

```

```

cfg = windows.load_manifest("run.yaml")
assert windows.is_right_open(cfg.windows.base)
assert windows.is_mece(cfg.windows.base)
assert windows.covers_target(cfg.windows.base, cfg.definable)
windows.log_partition(cfg, out="logs/windows_partition.txt")

```

T.8. δ -aggregation (per window, per degree, per τ)

Aggregation must respect the declared quantale V (Appendix S) and the canonical channels. Layer tags refine δ^{alg} ; advisory terms remain δ^{spec} .

```

# scripts/ledger_aggregate.py (pseudocode; Appendix S semantics)
from akhdpst import ledger, quantale

```

```

cfg = ledger.load_manifest("run.yaml")
V = quantale.load(cfg.quantale)

```

```

L = ledger.load("logs/ledger.json")

```

```

for (W,i,tau) in L.frames():
    # layer tags (optional)
    dTr = L.get("delta^Tr", W,i,tau, default=V.zero())
    dFun = L.get("delta^Fun", W,i,tau, default=V.zero())
    dGal = L.get("delta^Gal", W,i,tau, default=V.zero())

    # canonical channels
    dDisc = L.get("delta^disc", W,i,tau, default=V.zero())
    dMeas = L.get("delta^meas", W,i,tau, default=V.zero())
    dSpec = L.get("delta^spec", W,i,tau, default=V.zero())

    # algebraic total (refinement convention)
    dAlg = V.op(dTr, V.op(dFun, dGal))

    # window total
    dTot = V.op(dAlg, V.op(dDisc, V.op(dMeas, dSpec)))
    L.store("delta_total", W,i,tau, dTot)

```

```
ledger.write(L, "logs/ledger_aggregated.json")
```

T.9. Notebook cell skeletons (per stage; after-collapse inputs only)

Each notebook cell is required to: (i) read `run.yaml` as the source of truth, (ii) produce cache artifacts indexed by (W, i, τ) , (iii) emit ledger entries in V using canonical keys.

T.9.1. 01_stage_tropical.ipynb

```
cfg = load_yaml("run.yaml")
W,i,tau = "W1", 1, cfg.tau.grid.start
trop = tropical_readout(W,i,tau,cfg)
ledger.add("delta^Tr", W,i,tau, trop.delta_Tr)
save_json(trop, f"cache/tropical_{W}_{i}_{tau}.json")
```

T.9.2. 02_stage_lmhs.ipynb (advisory only)

```
trop = load_json(f"cache/tropical_{W}_{i}_{tau}.json")
lmhs = lmhs_proxy(trop, cfg.lmhs.proxies)
ledger.add("delta^spec", W,i,tau, lmhs.delta_spec) # may be zero by policy
save_json(lmhs, f"cache/lmhs_{W}_{i}_{tau}.json")
```

T.9.3. 03_stage_padic.ipynb

```
pad = padic_readout(W,i,tau, field=cfg.padic.field, p=cfg.padic.p)
ledger.add("delta^Fun", W,i,tau, pad.delta_Fun)
save_json(pad, f"cache/padic_{W}_{i}_{tau}.json")
```

T.9.4. 04_stage_arithmetic.ipynb (control / overlap)

```
ctl = control_bounds(W,i,tau,cfg) # Appendix R
ledger.add("delta^Gal", W,i,tau, ctl.delta_Gal)
PiT = read_Pi_then_Ttau(W,i,tau,cfg) # after-collapse object
E1 = energy_bins_after_collapse(PiT,cfg)
if E1.is_zero(): bridge_certify_after_collapse(PiT,W,i,tau,cfg)
```

T.9.5. 05_convergence_manager.ipynb

```
info = iterate_until_stable(W,i,tau0=cfg.tau.grid.start)
save_json(info, f"cache/convergence_{W}_{i}.json")
```

T.9.6. 06_gate_cascade.ipynb

```
res = run_cascade(W,i,tau,cfg)
save_json(res, f"cache/gate_{W}_{i}_{tau}.json")
```

T.9.7. 07_counterexample_hunter.ipynb

```
cases = hunt_all(W,i,tau,cfg)
save_json(cases, f"cache/hunter_{W}_{i}_{tau}.json")
```

T.10. CI/test integration (Chapter 12)

tests:

- name: "T_Vshift"
script: "pytest tests/test_vshift.py"
- name: "T_Definable"
script: "python scripts/mece_check.py"
- name: "T_Iwasawa"
script: "pytest tests/test_control_overlap.py"
- name: "T_GateCascade"
script: "pytest tests/test_gate_cascade.py"
- name: "T_Convergence"
script: "pytest tests/test_convergence_manager.py"
- name: "T_Hunter"
script: "pytest tests/test_hunter_regressions.py"
- name: "Demo_GL1"
script: "python scripts/demo_gl1.py"
- name: "Demo_GL2"
script: "python scripts/demo_gl2.py"

artifacts:

- "logs/ledger.json"
- "logs/ledger_aggregated.json"
- "logs/windows_partition.txt"
- "cache/convergence_W1_1.json"

T.11. GL(1) / GL(2) minimal workflows (non-gating demos; [Spec])

These demos are *integration* checks for the pipeline contracts (windowing, ledger, gates). They are explicitly not claims of number-theoretic correctness beyond logged budgets and after-collapse tests.

T.11.1. GL(1) (cyclotomic class-module *surrogate*).

```
# scripts/demo_gl1.py (pseudocode)
from akhdpst import pipeline

cfg = pipeline.load_manifest("run.yaml")
W,i,tau = "W1", 1, cfg.tau.grid.start

ctl = pipeline.demo.gl1_control_surrogate(cfg.demo.gl1)
pipeline.ledger.add("delta^Gal", W,i,tau, ctl.delta_Gal)

PiT = pipeline.read_Pi_then_Ttau(W,i,tau,cfg)
gap = pipeline.gap_tau(PiT,i,tau,cfg)
SigmaDelta = pipeline.ledger.window_total(W,i,tau,cfg)

print("GL(1) demo:", "PASS" if SigmaDelta < gap else "FAIL")
```

T.11.2. GL(2) (toy elliptic curve workflow; [Spec]).

```

# scripts/demo_gl2.py (pseudocode)
from akhdpst import pipeline

cfg = pipeline.load_manifest("run.yaml")
W,i,tau = "W1", 1, cfg.tau.grid.start

sel = pipeline.demo.gl2_selmer_surrogate(cfg.demo.gl2)
pipeline.ledger.add("delta^Gal", W,i,tau, sel.delta_Gal)

PiT = pipeline.read_Pi_then_Ttau(W,i,tau,cfg)
gap = pipeline.gap_tau(PiT,i,tau,cfg)
 $\Sigma\delta$  = pipeline.ledger.window_total(W,i,tau,cfg)

print("GL(2) demo:", "PASS" if  $\Sigma\delta < \text{gap}$  else "FAIL")

```

T.12. Invariants enforced by templates (audit contract)

- **After-collapse only.** No comparator reads pre-collapse metrics; all distances are computed on $\mathbf{T}_\tau \mathbf{P}_i$.
- **Definable finiteness.** All loops iterate over finitely many definable cells/events on bounded windows (Appendix H/J/Q).
- **Quantale compliance.** Ledger aggregation uses the declared \oplus in V with optional, logged change-of-base morphisms (Appendix S).
- **Gate separation.** LMHS/tropical proxies never gate alone; they can only trigger certified after-collapse tests (Appendix P).
- **Control accounting.** Finite kernel/cokernel contributions are embedded via η and recorded as algebraic budgets (Appendix R/S).
- **Restart/Summability.** Convergence Manager applies Appendix J (Restart + P6) and records stability bands *as data* (Appendix J/M).

T.13. Minimal run example (end-to-end; reproducible)

```

# 1) Fill run.yaml per T.2 and choose quantale per Appendix S.
# 2) Validate MECE coverage:
python scripts/mece_check.py
# 3) Execute  $\tau$ -sweep (cache + ledger):
python scripts/tau_sweep.py
# 4) Aggregate ledger totals in V:
python scripts/ledger_aggregate.py
# 5) Convergence Manager:
python scripts/convergence_manager.py
# 6) Gate Cascade per (W,i,tau):
python scripts/gate_cascade.py
# 7) (Optional) Counterexample Hunter:
python scripts/hunter_generate.py
# 8) Run CI tests (Chapter 12):
pytest -q

```

T.14. Non-claims (scope guard)

These templates assert no pre-collapse monotonicity, no commutation beyond recorded 2-cell bounds, no global $\text{PH}_1 \Leftrightarrow \text{Ext}^1$, and no analytic uniformity beyond logged δ -budgets. All guarantees are confined to bounded definable windows with after-collapse measurements, deletion-type-only monotonicity, and the quantale semantics of Appendix S.

Appendix U. AI Agent Specifications (Hunter / Mapper / Lifter) [Spec]

U.0. Purpose and relation to Part II (Spec-only, auditable search)

This appendix specifies the operational semantics of the AI agents introduced in Part II (Chapters 11–11). All statements herein are classified as **[Spec]**: they define a *reproducible, auditable search-and-assembly protocol* for producing candidate Proof Objects (Part II), without extending the proven collapse theorems of Part I.

Canon constraints (mandatory). Throughout Part II, all quantitative decisions are evaluated *after collapse* in the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare in Pers}_k^{\text{cons}}}.$$

All time windows are right-open and MECE on the target domain (App. G/H/J/Q). No pre-collapse comparison is in scope.

δ -ledger discipline (mandatory). All defects/budgets are recorded in a fixed commutative quantale V (App. S) via the δ -ledger (App. L/S/T) with the canonical decomposition

$$\delta = \delta^{\text{alg}} \oplus \delta^{\text{disc}} \oplus \delta^{\text{meas}} \quad (\text{canon; App. L/S}).$$

Advisory/heuristic uncertainty introduced by agents (e.g. gradient estimates, surrogate models) must be recorded as a *tagged* contribution within δ^{meas} (default) or δ^{disc} when appropriate, and may additionally be reported under a non-gating label δ^{spec} for transparency. Layer tags may refine δ^{alg} (e.g. δ^{Gal} , δ^{Tr} , δ^{Fun} , δ^{lift} , App. G/L/S/T).

Proposer–Verifier separation (mandatory). Agents in Part II act only as *Proposers*. All certified acceptance/rejection is determined solely by after-collapse Gates/tests and their logged budgets (Ch. 11/12; App. J/N/R/T). Advisory modules (e.g. tropical/LMHS proxies) are *never* Gates.

Remark R.1 (Terminology: terrain cells vs. time windows). To avoid collisions:

- A *Terrain Cell* is a definable subset $C_\alpha \subset \mathcal{M}$ of the *parameter space* \mathcal{M} navigated by agents (Part II).
- A *time window* is a right-open definable subset $W \subset \mathbb{R}$ (o-minimal) or $W \subset \text{VG}$ (Denef–Pas) used for after-collapse persistence comparisons (App. H/J/Q).

All Gate checks are evaluated on time windows W (after collapse), while the Mapper assembles coverage over terrain cells C_α .

U.1. Agent taxonomy, state, and shared objects

Definition R.2 (Agent types and roles). We distinguish three classes of autonomous agents with distinct responsibilities:

- **Hunter (H):** local navigator/optimizer on the parameter space \mathcal{M} . It explores within a Terrain Cell C_α to reduce a scalar *Defect Potential* Φ_τ computed from *after-collapse* ledger/test readouts.
- **Mapper (M):** global assembler. It verifies local certificates produced by Hunters and stitches them into a Coverage Graph via Overlap Gates.
- **Lifter (L):** singularity handler. It is invoked when a Hunter encounters a Type IV obstruction (diagnostics $(\mu, u) \text{ eq}(0, 0)$, App. D) and attempts a controlled dimensional extension subject to a *lifting penalty* recorded in the δ -ledger.

Definition R.3 (Shared manifest and reproducibility anchors). All agents operate under a single manifest `run.yaml` (App. G) which pins: (i) the quantale V and scalarization policy π (App. S), (ii) the time-window partition (MECE, right-open) and definability mode (App. H/J/Q), (iii) the τ -policy and degree list $\{i\}$, (iv) Gate order and enabled tests (Ch. 12; App. J/N/R/T), (v) deterministic numeric policy and serialization policy (App. G). Any agent output is valid as a Proof Object component only if it references the manifest hash (`run_yaml_hash` / content hash, App. G).

Definition R.4 (Defect Potential Φ_τ (scalarization; Spec)). Fix a monotone scalarization $\pi : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ logged in `run.yaml` (App. G/S). For a parameter point $x \in \mathcal{M}$, define $\Phi_\tau(x)$ from after-collapse windowwise budgets as

$$\Phi_\tau(x) := \max_{(W,i) \in \mathcal{W}_{\text{time}} \times I} \left(\pi(\Sigma \delta_W(i; x)) - \pi(\text{gap}_\tau(i; W, x)) \right)_+,$$

where $\Sigma \delta_W(i; x) \in V$ is the ledger total on (W, i) at x (App. J/S/T), $\text{gap}_\tau(i; W, x) \in V$ is the B-Gate⁺ threshold (App. J), and $\mathcal{W}_{\text{time}}$ denotes the finite MECE window family declared in `run.yaml`. Thus $\Phi_\tau(x) = 0$ implies that all required windowwise B-Gate⁺ inequalities hold (after collapse).

Definition R.5 (Gradient estimates are Spec objects). Any occurrence of a “gradient” in Part II means a *logged, reproducible estimate* $\widehat{\nabla} \Phi_\tau(x)$ produced by the Gradient Oracle specified in Chapter 13 (Spec. 11.8) under the manifest block `grad_policy` (App. G). Unlogged gradients are invalid and must not be used for any decision trace.

Definition R.6 (Hunter state tuple). A Hunter at step k maintains the state

$$S_k^H := (x_k, C_k, \Phi_\tau(x_k), \widehat{g}_k, \text{ctx}_k),$$

where $x_k \in \mathcal{M}$, C_k is the containing Terrain Cell, $\Phi_\tau(x_k)$ is as in Def. R.4, \widehat{g}_k is an *advisory* gradient estimate or surrogate direction (Def. R.5), and ctx_k contains the pinned (τ, i) -bundle, window IDs, and manifest hash.

Definition R.7 (Mapper state tuple). The Mapper maintains

$$S^M := (\mathcal{T}_{\text{valid}}, \mathcal{G}, \text{ctx}),$$

where $\mathcal{T}_{\text{valid}}$ is the set of Terrain Cells C_α with verified certificates, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the Coverage Graph with $\mathcal{V} = \mathcal{T}_{\text{valid}}$, and ctx pins the manifest hash and test policy.

Definition R.8 (Lifter state tuple). A Lifter instantiated at a singularity maintains

$$S^L := (x_{\text{sing}}, C_{\text{sing}}, \Phi_\tau(x_{\text{sing}}), (\mu, u), \ell, \text{ctx}),$$

where (μ, u) are the after-collapse tower diagnostics (App. D), $\ell \in \mathbb{Z}_{\geq 0}$ is the lifting depth (number of auxiliary axes introduced), and ctx includes the manifest hash and the current ledger snapshot ID.

U.2. Hunter protocol (operational semantics; Spec)

Specification R.9 (Hunter action semantics). At any state S_k^H , the Hunter must select exactly one action from the list below. *All certified decisions are based only on after-collapse readouts* ($\mathbf{T}_\tau \mathbf{P}_i$) and logged ledger totals (App. J/S/T). Advisory computations must be fully logged (Sec. R).

- **grad_estimate**: Query the Gradient Oracle (Spec. 11.8) at x_k and produce $\widehat{g}_k = \widehat{\nabla} \Phi_\tau(x_k)$, together with an estimated variance (or certified upper bound). The estimation error must be charged to the δ -ledger (default: δ^{meas} with tag δ^{spec}). This action is advisory and cannot certify validity.
- **gradient_step**: If in a *Ridge* regime (heuristically, $\Phi_\tau(x_k) > 0$ and a usable estimate $\widehat{g}_k \neq 0$ exists), propose an update $x_{k+1} = \text{Update}(x_k, \widehat{g}_k)$ constrained to remain inside C_k . The proposal must log step size, projection/clipping to C_k , and the acceptance/rejection rule. This action is advisory.
- **restart**: If descent stagnates (e.g. $\|\widehat{g}_k\|$ below a logged tolerance) while $\Phi_\tau(x_k) > 0$, invoke Restart Logic (App. J; App. T) to refine the current Terrain Cell C_k into subcells $\{C'\}$ (definable refinement; finite subdivision policy logged). Restart is a search operation and does not create certificates.
- **validate**: If in a *Plain* regime ($\Phi_\tau(x_k) = 0$), request a certified Gate Cascade on the relevant time-window bundle (W, i, τ) (App. T):

$$\text{B-Gate}^+ \rightarrow \text{PF/BC} \rightarrow \text{Overlap Gate}.$$

If the cascade passes under the declared budgets, then the containing Terrain Cell C_k may be flagged valid and packaged as a Local Certificate for the Mapper (Def. R.10).

- **escalate**: If in a *Peak* regime (e.g. $\Phi_\tau(x_k) \geq \lambda_{\text{sing}}$ for a logged threshold) *or* if Type IV diagnostics are detected, i.e.

$$(\mu, u) \neq (0, 0) \quad (\text{Type IV; App. D; canon: nonzero means obstruction}),$$

halt local navigation and invoke the Lifter (Sec. R).

Logging discipline. Every executed action appends a linked entry to the Action Log (Sec. R) including: manifest hash, RNG seed/state, (W, i, τ) -bundle IDs, ledger snapshot IDs, diagnostics (μ, u) when available, and hashes of referenced artifacts/certificates (App. G). For **grad_estimate**, the tuple (method, stencil, seed, variance) is mandatory.

Definition R.10 (Local certificate (Hunter output; auditable)). A Local Certificate attached to a Terrain Cell C_α consists of:

1. the manifest hash and time-window partition IDs (App. G),
2. the list of (W, i, τ) checks performed,
3. Gate Cascade results (pass/fail per stage) and the recorded δ -ledger totals $\Sigma \delta_W(i) \in V$,
4. the diagnostics (μ, u) and Type IV flags (App. D),
5. cryptographic hashes of artifacts used by checks (bars/spec/ext/phi/Lambda_len; App. G),
6. a deterministic replay recipe pointer (container digest / code hash; App. G).

Advisory outputs (LMHS/tropical proposals) may be included only as tagged δ^{meas} contributions (and optionally reported as δ^{spec}) and never as Gates.

U.3. Mapper protocol (Coverage Graph and assembly)

Specification R.11 (Mapper update rule). Upon receiving a Local Certificate for a Terrain Cell C_α , the Mapper must:

1. **Verify certificate integrity.** Check the manifest hash match, required test set presence (Ch. 12), and ledger completeness for canonical channels (App. L/S/T). If any mandatory component is missing, reject the update.
2. **Vertex insertion.** Insert C_α into \mathcal{V} and into $\mathcal{T}_{\text{valid}}$.
3. **Overlap verification.** For each existing $C_\beta \in \mathcal{T}_{\text{valid}}$ with $C_\alpha \cap C_\beta \neq \emptyset$, run the certified Overlap Gate on the overlap region, evaluated only after collapse on the time-window bundle:
 - (a) restrict parameters to $C_\alpha \cap C_\beta$ and evaluate each time window W ;
 - (b) compute/read $\mathbf{T}_\tau \mathbf{P}_i(\cdot)$ objects on overlaps;
 - (c) run the certified Overlap Gate (App. R/N/J; residuals recorded in the δ -ledger);
 - (d) if successful, insert an undirected edge $(\alpha, \beta) \in \mathcal{E}$.

Definable finiteness (finite events, finite Čech depth) must be logged under definable (App. G/H/J/Q).

4. **Global coverage check.** Periodically verify whether the union of Terrain Cells in the largest connected component of \mathcal{G} covers the target domain $\mathcal{M}_{\text{target}}$ specified by the Proof Object. The coverage criterion and any tolerated uncovered residue must be explicit and logged.

U.4. Lifter protocol (dimension management; Type IV handling)

Specification R.12 (Lifter operational semantics). When invoked at S^L , the Lifter attempts to resolve a Type IV obstruction by extending the parameter space. All lifting steps are **[Spec]** and must preserve the canon constraints: after-collapse evaluation, ledger accounting in V , and definable finiteness of the induced subproblems (App. G/H/J/Q/S/T).

1. **Axis selection (finite catalog).** Select one or more auxiliary axes \mathcal{A}_j from the finite axis catalog pinned in `run.yaml` (implementation notes; App. T), e.g. smoothing width, truncation order, arithmetic level, or model hyperparameters. All axes and admissible ranges must be declared in the manifest; axes are not allowed to modify Core claims.
2. **Directional test (advisory).** Evaluate an advisory directional decrease test for Φ_τ along \mathcal{A}_j at $(x_{\text{sing}}, 0)$ in $\mathcal{M} \times \mathcal{A}_j$, using a deterministic finite-difference rule pinned by `grad_policy` (App. G; Spec. 11.8).
3. **Augmented gap condition (certified budget check).** A lift to depth $\ell + 1$ introduces a lifting penalty $\delta^{\text{lift}}(\ell + 1) \in V$, recorded as a refinement tag of δ^{alg} (App. L/S/T). The lift is permitted only if, on each required time window and degree,

$$\Sigma \delta_W(i; x_{\text{sing}}) \oplus \delta^{\text{lift}}(\ell + 1) < \text{gap}_\tau(i; W, x_{\text{sing}}),$$

i.e. the B-Gate⁺ inequality remains satisfied after charging the lift penalty (App. J/S).

4. **Commit or fail.**

- **Commit:** If the directional test indicates decrease for some axis \mathcal{A}_j and the Augmented gap condition holds, commit the lift, charge $\delta^{\text{lift}}(\ell + 1)$, and spawn a new Hunter on $\mathcal{M}^{(\ell+1)} := \mathcal{M}^{(\ell)} \times \mathcal{A}_j$, with a new Terrain Cell partition (definable refinement policy logged).
- **Fail / Terminal Barrier (candidate counterexample region):** Otherwise, mark $(x_{\text{sing}}, C_{\text{sing}})$ as a Terminal Barrier and halt escalation. This yields a counterexample *candidate record* (not a theorem): diagnostics, ledger traces, and a reproducible failure-mode summary.

Mandatory lift trace. Every lift attempt must append a `lift_attempt` entry to the Action Log including: chosen axis, axis value(s), directional test output, penalty δ^{lift} , gap check pass/fail, and an explicit commit/reject rationale (Sec. R).

U.5. Action log schema (JSON/YAML; mandatory, linked, replayable)

Definition R.13 (Action log entry). A single log entry Entry_k is the tuple

$$\text{Entry}_k = (k, \text{ts}, \text{agent}, \text{seed}, \text{manifest}, x_k, C_k, \Phi_\tau(x_k), A_k, \text{aux}, \text{hash}),$$

where k is a step index, ts a timestamp (optional), agent encodes agent ID/type, seed encodes the RNG state, manifest is the manifest hash, $x_k, C_k, \Phi_\tau(x_k)$ are the state components, A_k is the action label, aux stores action-specific structured data, and hash is a cryptographic digest of the canonical serialization of the entry (App. G).

Specification R.14 (Schema requirements (canonical keys)). A concrete implementation (JSON Lines recommended) MUST encode the following keys.

Top-level keys (every entry).

```

k: <int>                # step index
ts: <string|null>        # optional timestamp
agent: { id: "...", type: "H|M|L" }
manifest: { run_id: "...", run_yaml_hash: "sha256:..." }
seed: { value: <int>, rng: "...", rng_state_hash: "sha256:..." }

state:
  x: <json>              # parameter point (typed payload)
  cell_id: "sha256:..." # Terrain Cell identifier (hash of definable description)
  Phi: <float>           # Phi_tau(x) (scalar)
bundle:
  tau: <float>
  degrees: [ ... ]       # list of i
  windows: [ ... ]       # list of window IDs (right-open; MECE)

ledger:
  snapshot_id: "sha256:..."
  totals:
    delta_alg: <float>
    delta_disc: <float>
    delta_meas: <float>
  tags:                  # optional refinements/tags

```

```

delta_Gal: <float|null>
delta_Tr: <float|null>
delta_Fun: <float|null>
delta_lift: <float|null>
delta_spec: <float|null>      # optional transparency tag (non-gating)

diagnostics:
  mu: <int|null>
  nu: <int|null>
  type_iv: <bool|null>

action: "grad_estimate|gradient_step|restart|validate|escalate|lift_attempt|map_update"
aux: { ... }                  # action-specific payload (see below)

integrity:
  refs:
    bars: "sha256:..."      # optional, if referenced in this step
    spec: "sha256:..."
    ext: "sha256:..."
    phi: "sha256:..."
    Lambda_len: "sha256:..."
    prev_hash: "sha256:..." # optional hash-chain
  hash: "sha256:..."        # hash of canonical serialization of this entry

```

Action-specific payloads (mandatory when action occurs). (1) `grad_estimate`. The following keys are mandatory:

```

aux:
  grad:
    method: "finite_difference|SPSA|surrogate"
    norm: "fro|op|custom"
    stencil: { kind: "one_sided|two_sided", eps: <float>, coords: "all|subset:[...]" }
    seed: <int>
    eval_count: <int>
    variance: <float>          # estimate or certified upper bound
    variance_mode: "estimate|upper_bound"
    delta_charge:
      channel: "delta_meas|delta_disc"
      amount: <float>

```

The tuple (method, stencil, seed, variance) is required for replayability.

(2) `gradient_step`.

```

aux:
  step:
    step_size: <float>
    rule: "projected_descent|trust_region|other"
    accept: <bool>
    reason: "<string>"

```

(3) restart.

```
aux:
  restart:
    criterion: "<string>"
    refined_cells: ["sha256:...", ...]
    refinement_policy: "<string>"    # definable finite subdivision description
```

(4) validate.

```
aux:
  gates:
    b_gate_plus: { passed: <bool>, slack: <float|null> }
    pfbc:        { passed: <bool|null>, residual: <float|null> }
    overlap:     { passed: <bool|null> }
    reason: "<string>"
```

(5) lift_attempt. (Lifter trace; mandatory when lifting is invoked)

```
aux:
  lift:
    depth_from: <int>
    depth_to: <int>
    axis: "<string>"          # from finite axis catalog
    axis_value: <json>
    directional_test: { value: <float>, passed: <bool> } # advisory
    penalty:
      delta_lift: <float>
      charged_to: "delta_alg"
    augmented_gap_check: { passed: <bool>, slack: <float|null> }
    commit: <bool>
    rationale: "<string>"
```

(6) map_update. (Mapper event)

```
aux:
  mapper:
    received_cell: "sha256:..."
    accepted: <bool>
    overlap_edges_added: [["sha256:...", "sha256:..."], ...]
    reason: "<string>"
```

Replayability condition (operational). A third party providing the same manifest hash, initial seed/state, and running the trusted AK Core (Part I) with the declared deterministic numeric policy must reproduce identical certificates and identical serialized artifacts referenced by the log, up to the declared serialization and tolerance policy (App. G). Any missing mandatory key (notably `grad.method/stencil/seed/variance` when gradients are used, or the lift payload when lifting occurs) invalidates the corresponding agent-based claim.

U.6. Proof Object packaging (Spec; what counts as valid output)

Definition R.15 (Map-of-Validity Proof Object (Part II; Spec)). A Part II Proof Object consists of:

1. the manifest `run.yaml` and its hash (App. G),
2. the set of validated Terrain Cells $\mathcal{T}_{\text{valid}}$ with Local Certificates (Def. R.10),
3. the Coverage Graph \mathcal{G} with Overlap Gate edge evidence (Sec. R),
4. the complete Action Log for all agents (Sec. R),
5. the ledger exports and aggregated totals used in B-Gate⁺ and PF/BC/Overlap checks (App. J/N/R/S/T),
6. any Terminal Barrier records (if present), as counterexample candidates only.

Advisory components (LMHS/tropical suggestions) may be included only as tagged δ^{meas} contributions (and optionally reported as δ^{spec}) and never as stand-alone validators.

U.7. Non-claims (scope guard)

No statement herein asserts new pre-collapse control, global commutation beyond recorded 2-cell bounds, or any converse beyond the one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ in $D^b(k\text{-mod})$ (Ch. 3; App. C). The agents do not create mathematical truth; they create auditable search traces whose certified components are limited to after-collapse Gates/tests under the UCC and the δ -ledger discipline (Ch. 1/2; App. J/N/R/S/T).

Appendix V. Validity Map Formalism [Spec; stratified semantics and Global Certificate kernel]

Standing conventions (canon). All quantitative evaluations are *after collapse* in the fixed order

$$\boxed{\text{for each } t \Rightarrow \mathbf{P}_i \Rightarrow \mathbf{T}_\tau \Rightarrow \text{compare on Pers}^{\text{ft}}} \quad (\text{Ch. 1/2; App. J/S/T}).$$

Windows on the time axis are definable and right-open, and the chosen cover has finite event counts and finite Čech depth on bounded ranges (App. H/J/Q). All defects/budgets are written to a fixed commutative quantale V via the δ -ledger (App. S), aggregated by \oplus . Type IV obstruction means $(\mu, u) \text{ eq}(0, 0)$ (App. D; canon). PF/BC and arithmetic control (Overlap Gate) are evaluated only after collapse (App. N/R). All material below is [Spec] unless explicitly stated otherwise; it defines operational proof objects and does not extend proven theorems of Part I.

V.0. Purpose and scope

This appendix formalizes the *Map of Validity* introduced in Chapter 11. It defines (i) the validity stratification of the parameter space \mathcal{M} via the Defect Potential Φ_τ , and (ii) the *Global Certificate* data structure whose verification reduces global acceptance to finite, auditable checks under the Unified Collapse Contract (UCC; Ch. 1).

V.1. Level-set topology of Φ_τ

Definable parameter space. Let \mathcal{M} be a definable topological space (e.g. definable in an o-minimal expansion $\mathbb{R}_{\text{an}, \text{exp}}$, or definable in a Denef–Pas structure; see App. H/Q for definability conventions). A *Terrain Cell decomposition* \mathcal{T} is a finite or locally finite definable cover of \mathcal{M} by definable sets (Terrain Cells) with logged overlap multiplicity and (when needed) a refinement policy (App. U/T).

Definition S.1 (Defect Potential (after-collapse scalarization)). Fix: (i) a collapse scale $\tau > 0$, (ii) a finite index set of degrees I , (iii) a bundle of time windows $\mathcal{W}_{\text{time}}$ (right-open, definable; App. H/J/Q), (iv) a quantale V with ledger aggregation \oplus (App. S), and (v) a monotone scalarization $\pi : V \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ logged in `run.yaml` (App. S/U/T). For $x \in \mathcal{M}$, define $\Phi_\tau(x) \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ by

$$\Phi_\tau(x) := \max_{(W,i) \in \mathcal{W}_{\text{time}} \times I} \left(\pi(\Sigma \delta_W(i; x)) - \pi(\text{gap}_\tau(i; W, x)) \right)_+,$$

where $\Sigma \delta_W(i; x) \in V$ is the windowwise ledger total (App. J/S/T) and $\text{gap}_\tau(i; W, x) \in V$ is the B-Gate⁺ threshold (App. J). Thus $\Phi_\tau(x) = 0$ means that the required after-collapse inequalities pass on all declared windows/degrees.

Definition S.2 (Operational strata). Fix safety thresholds gap_τ (certification limit) and λ_{sing} (singularity threshold), both pinned by policy and logged (App. J/U/T). Define:

1. **Valid set:**

$$Z_{\text{Valid}} := \{x \in \mathcal{M} \mid \Phi_\tau(x) = 0\}.$$

This is the region where the *after-collapse Gate Cascade* (App. T) succeeds under the declared δ -budgets.

2. **Noise set:**

$$Z_{\text{Noise}} := \{x \in \mathcal{M} \mid 0 < \Phi_\tau(x) < \lambda_{\text{sing}}\}.$$

This is the region of nonzero but non-singular defect potential (operational Types I–III). Hunter agents search here.

3. **Singular set:**

$$Z_{\text{Sing}} := \{x \in \mathcal{M} \mid \Phi_\tau(x) \geq \lambda_{\text{sing}} \text{ or } (\mu(x), u(x)) \text{ eq}(0, 0)\}.$$

This is the region of essential obstructions (Type IV by diagnostics and/or large potential; App. D/U).

Remark S.3 (About “closed/open” claims). We do not assume topological regularity of Φ_τ beyond what is logged and tested (e.g. definability, finite-event behavior on time windows, optional Lipschitz controls; App. H/J/Q/S). Consequently, openness/closedness of the strata is not asserted unless proven in the chosen definable regime and recorded as such.

Definition S.4 (Validity Map at scale τ). The *Validity Map* at scale τ is the tuple

$$\mathfrak{B}_\tau := (\mathcal{M}, \Phi_\tau, \mathcal{T}; Z_{\text{Valid}}, Z_{\text{Noise}}, Z_{\text{Sing}}),$$

where \mathcal{T} is a Terrain Cell decomposition compatible with the operational strata in the following sense: for each $\mathcal{W}_\alpha \in \mathcal{T}$, either (i) Φ_τ has constant regime on \mathcal{W}_α (all points certify / all are noise / all are singular), or (ii) Φ_τ has a logged control law on \mathcal{W}_α (e.g. a Lipschitz bound with respect to a declared metric) sufficient for the Mapper’s refinement/coverage logic (App. U/T).

Specification S.5 (Operational global acceptance condition (Spec)). Operationally, we accept *global validity* on a target domain $\mathcal{M}_{\text{target}} \subseteq \mathcal{M}$ only via a *Global Certificate* (Section S) that provides:

- **Coverage:** $\bigcup_{\alpha \in A} \mathcal{W}_\alpha \supseteq \mathcal{M}_{\text{target}}$, verified by a covering proof in the definable category used (o-minimal / Denef–Pas / explicit polyhedral cover).
- **Local validity:** for each vertex cell \mathcal{W}_α , the after-collapse Gate Cascade passes on the declared time-window bundle (B-Gate⁺ \rightarrow PF/BC \rightarrow Overlap Gate), under recorded δ -budgets (App. T; App. J/N/R).

- **No certified singularities:** $\mathcal{W}_\alpha \cap Z_{\text{Sing}} = \emptyset$ for all validated cells, certified by diagnostics $(\mu, u) = (0, 0)$ and the logged Type IV policy (App. D/U).

No deformation-retract or density criterion is asserted as a theorem here; any such strengthening is [Spec] and must be logged with explicit hypotheses and tests.

V.2. Global Certificate format

Design principle. A Global Certificate is a finite, auditable object: verification is a finite traversal plus finite ledger aggregation in V (App. S/J/T), using only after-collapse comparators.

Definition S.6 (Global Certificate structure). A *Global Certificate* is a directed acyclic graph (DAG)

$$C_{\text{global}} := \langle \mathcal{V}, \mathcal{E}, \mathcal{M}_{\text{meta}} \rangle$$

with:

- **Vertices \mathcal{V} :** a finite collection of validated Terrain Cells $\{\mathcal{W}_\alpha\}_{\alpha \in A}$. Each vertex stores:
 - a definable descriptor (formula, constraints, or code-hash-defined predicate) for \mathcal{W}_α ;
 - the declared time-window bundle $\mathcal{W}_{\text{time}, \alpha}$ and degree set I_α ;
 - a *local validity token* consisting of: (i) manifest hash, (ii) Gate Cascade pass results, (iii) hashes of referenced persistence artifacts, (iv) windowwise ledger totals $\Sigma \delta_W(i; \mathcal{W}_\alpha) \in V$, and (v) diagnostics (μ, u) with Type IV flag policy (App. D/U/T).
- **Edges \mathcal{E} :** verified overlaps. An edge $\alpha \rightarrow \beta$ exists only if:
 - $\mathcal{W}_\alpha \cap \mathcal{W}_\beta \neq \emptyset$, and
 - the *after-collapse Overlap Gate* passes on the overlap, with any finite kernel/cokernel parts recorded as δ^{alg} (App. R; PF/BC compatibility App. N; finite-depth glue App. J/Q).

The edge record stores the overlap descriptor and the Overlap Gate evidence (hashes + budgets).

- **Metadata $\mathcal{M}_{\text{meta}}$:**
 - **Covering proof:** evidence that $\bigcup_{\alpha \in A} \mathcal{W}_\alpha \supseteq \mathcal{M}_{\text{target}}$, including the definable regime, MECE/refinement policy, and any residue tolerance (App. G/U).
 - **Summability / finiteness:** evidence that the aggregation performed by the verifier is finite: bounded definable covers on time windows (finite event/Čech depth) and P6-type summability where applicable (App. J/Q/S). The verifier aggregates only finite \oplus -sums.
 - **Type IV clearance (within certified region):** evidence that for all vertices, $(\mu, u) = (0, 0)$ on the declared checks, hence $Z_{\text{Sing}} \cap \left(\bigcup_{\alpha \in A} \mathcal{W}_\alpha \right) = \emptyset$ *within the certified regime and policy*.

Specification S.7 (Verification protocol (trusted kernel)). To verify C_{global} , a trusted kernel performs:

1. **Manifest check.** Confirm the manifest hash and the enabled test set (Ch. 12) match all vertex/edge records.
2. **Local check (per vertex).** For each $\mathcal{W}_\alpha \in \mathcal{V}$, re-run the after-collapse Gate Cascade on the declared time windows:

$$\text{B-Gate}^+ \rightarrow \text{PF/BC} \rightarrow \text{Overlap Gate},$$

recomputing (or verifying by hash) the referenced persistence artifacts and re-aggregating the window-wise ledger totals $\Sigma\delta_W(i; \mathcal{W}_\alpha) \in V$ (App. S/T). Confirm that Type IV diagnostics satisfy $(\mu, u) = (0, 0)$ under the declared policy (App. D).

3. **Edge/overlap check.** For each $\alpha \rightarrow \beta \in \mathcal{E}$, re-verify the overlap descriptor and the after-collapse Overlap Gate evidence, including any control-derived finite kernel/cokernel budget terms in δ^{alg} (App. R).
4. **Coverage check.** Verify the covering proof that $\bigcup_{\alpha \in A} \mathcal{W}_\alpha \supseteq \mathcal{M}_{\text{target}}$, using the definable regime declared (o-minimal / Denef-Pas / explicit combinatorial cover).
5. **Global budget check.** Re-aggregate the declared global ledger total $\bigoplus_{\alpha \in A} \Sigma\delta(\mathcal{W}_\alpha)$ using the chosen quantale V , and confirm compliance with the global gap policy mandated by UCC (Ch. 1; App. J/S).

Under these checks, verification reduces to finite graph traversal plus finite aggregation in V .

V.3. Reproducibility hooks (run.yaml)

```

validity_map:
  tau: "<value>"
  phi:
    quantale: "<V_name>"
    scalarization_pi: "<pi: V -> R>=0>"
    sing_threshold: "<lambda_sing>"
    gap_policy: "<gap_tau policy>"
  target_domain: "<M_target descriptor>"
  terrain:
    definable_regime: "R_an,exp | Denef-Pas | explicit"
    cells: "<list or generator + hash>"
    refinement_policy: "<restart/refine rules>"
  global_certificate:
    format: "DAG"
  vertex_fields: ["cell_descriptor", "windows", "degrees", "gate_tokens", "ledger_totals", "diag_mu_nu", "hashes"]
  edge_fields: ["overlap_descriptor", "overlap_gate_token", "delta_alg_budget", "hashes"]
  checks:
    local_gate_cascade: true
    overlap_gate: true
    coverage_proof: true
    global_budget: true
  tests:
    T_GateCascade: true
    T_Definable: true
    T_Vsubadd: true

```

V.4. Summary

Appendix V formalizes the *output* of the AK-HDPST framework (Part II) as:

- a stratified semantics on \mathcal{M} induced by an after-collapse Defect Potential Φ_τ , and
- a finite, auditable Global Certificate DAG whose verification is a finite traversal plus quantale aggregation.

All acceptance statements are operational and confined to the declared target domain, the after-collapse Gate Cascade, and the logged δ -ledger discipline under UCC.

Appendix W. Bridge Programs — Detailed Specifications [Spec; Spectral-Gap guard-rails and B1/B2/B3 state machines]

Standing conventions (canon). All quantitative evaluations are *after collapse* in the fixed order

$$\boxed{\text{for each } t \implies \mathbf{P}_i \implies \mathbf{T}_\tau \implies \text{compare on Pers}^{\text{ft}}} \quad (\text{Ch. 1/2; App. J/S/T}).$$

Definability is o-minimal on the real side or Denef–Pas on the p -adic side (App. H/Q). Finite-event and finite Čech properties on bounded windows are assumed and must be logged (App. H/J/Q/T). All defect-s/budgets live in a fixed commutative quantale V and are aggregated by \oplus via the δ -ledger (App. S; entries $\delta^{\text{alg}}, \delta^{\text{disc}}, \delta^{\text{meas}}, \delta^{\text{spec}}, \delta^{\text{lift}}$). Type IV obstruction is *by canon* $(\mu, u) \text{ eq}(0, 0)$ (App. D); any appearance of Type IV triggers the escalation rules (App. U). This appendix is **[Spec]** throughout: it defines operational acceptance logic and logging discipline, and does not extend proven implications of Part I.

Notation warning. In Part II we use *Terrain Cells* $\mathcal{W}_\alpha \subset \mathcal{M}$ for parameter-space decomposition (App. U/V), and *time windows* $W \subset \mathbb{R}$ (or VG in Denef–Pas mode) for after-collapse persistence readouts (App. H/J/Q/T). Reverse-bridge checks in this appendix are *cell-local in \mathcal{M}* but evaluate $\mathbf{P}_i(\mathbf{T}_\tau \cdot)$ on the declared time-window bundle.

W.0. Purpose and scope

This appendix refines the Bridge Programs of Chapter 11 into executable specifications. Its role is *strictly guard-rail*: whenever an AI agent proposes a *reverse direction* ($\text{Ext}^1 = 0 \implies \text{PH}_1 = 0$) as part of a Proof Object, the AK Core accepts such a step *only* under the Spectral-Gap Condition and the state machines B1/B2/B3 below. The intent is to prevent numerical hallucination (false vanishing) and to force full auditability.

Non-goal: no global theorem $\text{Ext}^1 \Leftrightarrow \text{PH}_1$ is claimed. The only proven direction remains the one-way bridge $\text{PH}_1 \implies \text{Ext}^1$ in $D^b(k\text{-mod})$ under stated hypotheses (Ch. 3/11; canon).

W.1. Spectral-Gap Condition (SGC) as an audit gate

Policy boundary. Spectral indicators are *not* assumed to be filtered quasi-isomorphism invariants; they are admissible only as a logged audit channel with explicit policy (App. S/T; canon boundary in Ch. 2/5). Accordingly, the Spectral-Gap Condition is an **operational acceptance condition** and must be recorded as **[Spec]** with its computation policy, bounds, and sampling discipline.

Definition T.1 (After-collapse Laplacian profile (policy-driven)). Fix a Terrain Cell $\mathcal{W} \subset \mathcal{M}$, collapse scale τ , and a declared *realization policy* Pol_{Lap} (e.g. truncation order, discretization scheme, normalization, and any parameters such as $\beta, M(\tau), s$), logged in run.yaml (App. T/U/S). For each $x \in \mathcal{W}$, let $L_\tau(x) = L_\tau^{\text{Pol}_{\text{Lap}}}(x)$ be the *normalized combinatorial Laplacian* computed from the after-collapse object (i.e. from $\mathbf{P}_i(\mathbf{T}_\tau F_x)$ and the declared construction).

Define the **signal floor** and **noise ceiling** by:

1. **Signal floor** $\gamma_{\tau, \min}(\mathcal{W})$:

$$\gamma_{\tau, \min}(\mathcal{W}) := \inf_{x \in \mathcal{W}} \lambda_1^+(L_\tau(x)), \quad \lambda_1^+(L) := \min\{\lambda \in \sigma(L) \mid \lambda > 0\},$$

with the convention $\lambda_1^+(L) = +\infty$ if $\sigma(L) \subseteq \{0\}$ (everywhere-zero spectrum).

2. Noise ceiling $\delta_{\max}(\mathcal{W})$:

$$\delta_{\max}(\mathcal{W}) := \sup_{x \in \mathcal{W}} \pi(\Sigma\delta(x)),$$

where $\Sigma\delta(x) \in V$ is the aggregated ledger total relevant to the reverse-bridge attempt and $\pi : V \rightarrow \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is the declared scalarization (App. S/V).

Remark T.2 (How inf/sup are realized in the implementable range). On bounded definable windows, event finiteness and finite Čech depth reduce all required maxima to finite checks (App. H/J/Q). Operationally, the AK Core *does not accept* unverifiable analytic inf/sup: $\gamma_{\tau, \min}$ must be supplied as a certified lower bound and δ_{\max} as a certified upper bound, both produced from a finite cell/event decomposition or a logged finite sampling scheme with refinement guarantees. All such certificates are part of the Proof Object and hashed (App. U/T/V).

Specification T.3 (Spectral-Gap Condition $\text{SGC}(c)$). Fix a safety factor $c > 1$, logged per run. A Terrain Cell \mathcal{W} satisfies $\text{SGC}(c)$ at scale τ if

$$\gamma_{\tau, \min}(\mathcal{W}) > c \cdot \delta_{\max}(\mathcal{W}).$$

Default policy: enforce $c \geq 2$ unless a tighter factor is justified and recorded with explicit bounds and tests.

Remark T.4 (Interpretation). $\text{SGC}(c)$ enforces that the first nonzero spectral mode is separated from 0 by a margin larger than the worst admissible defect budget. This is used solely to block “near-zero” artifacts from being misread as true vanishing.

W.2. Program B1: Local reverse logic ($\text{Ext} \Rightarrow \text{PH}$) with guard-rails

Policy boundary (reverse direction). The reverse implication $\text{Ext}^1 = 0 \Rightarrow \text{PH}_1 = 0$ is **not** a theorem of the AK Core. Program B1 defines the *only* permissible way to produce a *Reverse Certificate* for a specific Terrain Cell, and such a certificate is valid only under the logged SGC policy, gate results, and ledger budgets.

Definition T.5 (B1 status codes). Program B1 returns one of:

Valid | Indeterminate | Reject | TypeIV | Barrier.

Valid means “reverse certificate issued under $\text{SGC}(c)$ and UCC guard-rails”. Indeterminate means “insufficient separation or insufficient bounds” (no reverse step allowed). Reject means “violated preconditions or failed audits”. TypeIV means “ $(\mu, u) \text{ eq}(0, 0)$ ” (must escalate; App. U). Barrier means “terminal obstruction candidate logged for B3”.

Specification T.6 (Program B1 (Reverse Bridge) — executable logic). **Inputs:** Terrain Cell $\mathcal{W} \subset \mathcal{M}$, scale τ , degree i (typically $i = 1$), declared time-window bundle $\mathcal{W}_{\text{time}}$, quantale V with scalarization π , SGC factor c , and realization policy Pol_{Lap} . All are read from `run.yaml` and hashed (App. T/U/V).

Procedure (must follow after-collapse order):

1. Precondition checks (hard gates).

- *Definability:* \mathcal{W} is definable and its descriptor validates (App. H/Q/U).
- *After-collapse Gate Cascade:* On \mathcal{W} , the declared Gate Cascade passes: B-Gate⁺ on $\mathbf{P}_i(\mathbf{T}_\tau \cdot)$, then PF/BC, then Overlap Gate on declared overlaps, with all finite parts logged in δ -ledger (App. T; App. J/N/R).

- *Type IV exclusion*: diagnostics satisfy $(\mu, u) = (0, 0)$ on \mathcal{W} (App. D/U). If $(\mu, u) \neq (0, 0)$, return TypeIV.
2. **Spectral audit (SGC)**. Compute or import certified bounds $\gamma_{\tau, \min}(\mathcal{W})$ and $\delta_{\max}(\mathcal{W})$ under the declared policies (Def. T.1, Rem. T.2). If $\text{SGC}(c)$ fails, return Indeterminate.
 3. **Observed Ext-bound (numerical/algorithmic check)**. Let $\epsilon_{\text{obs}}(\mathcal{W})$ be the observed magnitude of Ext^1 under the declared computation policy (e.g. cochain-level solver tolerances and truncations), logged as a budgeted measurement channel:

$$\epsilon_{\text{obs}}(\mathcal{W}) := \sup_{x \in \mathcal{W}} \left\| \text{Ext}^1(\mathcal{R}(\mathbf{T}_\tau F_x) |_{\mathcal{W}_{\text{time}}, k}) \right\|_{\text{obs}}.$$

Require the guarded inequality

$$\epsilon_{\text{obs}}(\mathcal{W}) \leq \delta_{\max}(\mathcal{W}).$$

If it fails, return Reject. (Operational meaning: the observed nonzero Ext is larger than what the ledger allows.)

4. **Issue Reverse Certificate (local, conditional)**. If all checks pass, return Valid and output a **Reverse Certificate** token stating:

On \mathcal{W} , under the declared realization policy Pol_{Lap} , $\text{SGC}(c)$, the after-collapse Gate Cascade, and the recorded δ -ledger budgets, the reverse step $\text{Ext}^1 = 0 \Rightarrow \text{PH}_1 = 0$ is accepted as an operational inference.

The token MUST include hashes of: \mathcal{W} descriptor, τ , c , $\gamma_{\tau, \min}$ certificate, δ_{\max} certificate, and the referenced persistence artifacts (App. U/V).

W.3. Program B2/B3: Global orchestration (proof mode / disproof mode)

Specification T.7 (Program B2: Global Regularity (Proof Mode; Spec)). **Goal**: Construct a Global Certificate covering $\mathcal{M}_{\text{target}}$ (App. V), where each certified cell supports any required reverse steps only via Program B1.

State: a queue Q of regions to process, and a Coverage Graph \mathcal{G} managed by the Mapper (App. U/V).

Algorithm (operational):

1. Initialize $Q \leftarrow \{\mathcal{M}_{\text{target}}\}$, $\mathcal{G} \leftarrow \emptyset$.
2. While $Q \neq \emptyset$:
 - Pop a region R from Q . A Hunter proposes a definable Terrain Cell $\mathcal{W} \subseteq R$ with a logged action trace (App. U).
 - Run Program B1 on \mathcal{W} .
 - If Valid: add \mathcal{W} as a vertex; invoke Mapper overlap verification (after-collapse Overlap Gate) to add edges (App. U/R).
 - If Indeterminate: apply Restart/Refinement policy to split R or refine \mathcal{W} into subcells with finite-event guarantees and updated budgets (App. J/T/U).
 - If TypeIV: invoke the Lifter (App. U). If lift commits under the augmented gap and δ^{lift} budget, push lifted cells into Q ; otherwise mark as Barrier and switch to B3.
 - If Reject: refine or discard per policy; rejection reasons must be logged (App. U/T).

3. Terminate with **Global Regularity (operational)** only when:

- the largest connected component of \mathcal{G} covers $\mathcal{M}_{\text{target}}$ (covering proof), and
- global ledger aggregation satisfies the UCC gap policy and P6/Summability constraints as declared (App. J/V/S).

Export a Global Certificate DAG (App. V) including all B1 tokens and hashes.

Specification T.8 (Program B3: Counterexample Hunt (Disproof Mode; Spec)). **Goal:** Isolate an *essential* singularity as a Certified Counterexample Candidate, with exhaustive logged lift attempts.

Algorithm (operational):

1. **Search.** A Hunter maximizes Φ_τ on $\mathcal{M}_{\text{target}}$ (or on unresolved regions), producing a peak candidate x^* with full action log (App. U).
2. **Verify Type IV signature.** Confirm $(\mu(x^*), u(x^*)) \text{ eq}(0, 0)$ after collapse (App. D). If not, the candidate is not Type IV and must be returned to B2 refinement.
3. **Exhaust lifts (finite catalog).** Enumerate the finite auxiliary-axis catalog declared for the run (App. U/S; logged). For each axis \mathcal{A}_j , evaluate the directional change policy (e.g. finite-difference/adjoint) for $\partial_{\mathcal{A}_j} \Phi_\tau(x^*, 0)$, and estimate the Lifting Penalty $\delta^{\text{lift}}(k + 1)$ charged to the ledger. If for all \mathcal{A}_j either:

$$\partial_{\mathcal{A}_j} \Phi_\tau(x^*, 0) \geq 0 \quad \text{or} \quad \Sigma \delta(x^*) \oplus \delta^{\text{lift}}(k + 1) \not\leq \text{gap}_\tau,$$

then declare the point **Essential** and mark it as a **Terminal Barrier** (App. U).

4. **Output.** Return x^* together with: the full ledger snapshot, diagnostics (μ, u) , failed lift attempts (with hashes), and the precise policy bundle (quantale V , scalarization π , SGC factor c , realization policy Pol_{Lap}).

W.4. Reproducibility hooks (run.yaml / logs)

```
bridge_programs:
  enabled_reverse_bridge: true/false
  tau: "<value>"
  degree: 1
  time_windows: "<bundle-id>"
  phi_policy:
    quantale: "<V_name>"
    scalarization_pi: "<pi: V->R>=0>"
    gap_policy: "<gap_tau rule>"
    sing_threshold: "<lambda_sing>"
  spectral_gap:
    factor_c: 2.0
    laplacian_policy: "<Pol_Lap id>"
    gamma_cert: "<path/hash of lower-bound cert>"
    delta_max_cert: "<path/hash of upper-bound cert>"
  B1:
    require_gate_cascade: true
    require_diag_mu_nu_zero: true
    statuses: ["Valid", "Indeterminate", "Reject", "TypeIV", "Barrier"]
```

```

B2:
  queue_policy: "<refine/restart rules>"
  stop_condition: "coverage+budget"
B3:
  lift_catalog: "<axis list/hash>"
  max_lift_depth: "<kmax>"
logs:
  reverse_certificates: "logs/reverse_tokens.jsonl"
  bridge_audits: "logs/bridge_audits.json"
tests:
  T_GateCascade: true
  T_Vsubadd: true
  T_Definable: true

```

W.5. Non-claims

No statement here asserts a mathematical theorem $\text{Ext}^1 = 0 \Rightarrow \text{PH}_1 = 0$. $\text{SGC}(c)$ is an operational audit gate only, dependent on a declared Laplacian realization policy. No pre-collapse comparisons are permitted. No spectral invariance beyond what is logged and tested is assumed. All reverse certificates are cell-local, policy-local, and invalid outside the logged run configuration.

W.6. Integration points

This appendix is consumed by Chapter 11 (Bridge Programs), Appendix U (agent protocols and logging), Appendix V (Global Certificate format), Appendix T (notebook/CI skeletons), Appendix S (quantale/scalarization policy), and Appendix J (Restart/Summability constraints). All references to overlaps and finite kernel/cokernel accounting use App. R, and PF/BC checks use App. N, always after collapse.

Appendix NS: Case Study [Spec] — 3D Incompressible Navier–Stokes (Exploration Mode)

NS.0. Status, boundary, and canon

Status (strict). This appendix is [Spec] and belongs to the *Exploration/Navigation* layer. It provides an auditable instantiation of the Part II machinery (windows, ledger, gates, Hunter/Mapper/Lifter) to the 3D incompressible Navier–Stokes equations (NSE). **No Millennium-claim is made.** Any output is empirical/certificational evidence *within declared search classes* and *within declared policies*. Nothing here upgrades or modifies the theorem-level core of Part I.

Canon alignment (v17.0). All constraints below are mandatory for consistency with v17.0:

- **After-collapse only.** Every metric that can influence a decision (gate pass/fail, regime label, certificate) is computed on the B-side: $\mathbf{T}_\tau \mathbf{P}_i(F)$. Pre-collapse readings may be recorded, but never gate.
- **Type IV meaning is fixed.** “Normal” means $(\mu, u) = (0, 0)$. **Type IV obstruction** means $(\mu, u) \text{ eq}(0, 0)$, computed *after* \mathbf{T}_τ under the declared tower policy.
- **Quantale ledger.** All budget accounting is performed in a declared commutative quantale V , with a declared scalarization/norm used *only* for reporting.

- **Proxy separation.** Tropical/LMHS/spectral proxies (if used) are advisory only and never gate alone.
- **Arithmetic overlap rule.** Arithmetic control participates only through the *Control* \Rightarrow *Overlap Gate*; finite parts are logged as δ^{alg} .
- **Gate order (after-collapse).** The Gate Cascade is

$$\text{B-Gate}^+ \longrightarrow \text{PF/BC} \longrightarrow \text{Overlap Gate}.$$

- **Restart/Summability.** The Convergence Manager enforces Restart/Summability windowwise and records stability bands.
- **Definable finiteness.** Time windows and event partitions are definable, so all enumerations are finite on bounded windows.

Non-claims (hard boundary). This appendix makes **no** claim of global regularity, no claim of equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$, and no claim that any chosen realization is canonical beyond the explicitly declared policy. In particular, any reverse-audit style step is prohibited unless guarded exactly as specified in the Bridge Programs layer (if invoked at all), and even then remains **[Spec]**.

NS.1. PDE arena and declared search class

Equation. We consider 3D incompressible NSE on \mathbb{T}^3 (or a declared bounded domain with periodic surrogate):

$$\partial_t u + (u \cdot \nabla)u + \nabla p = \nu \Delta u, \quad \nabla \cdot u = 0,$$

with viscosity $\nu > 0$ and initial data $u(0) = u_0$.

Regularity class and blow-up locus (classical). Fix $s > \frac{5}{2}$ and define the phase space

$$M_{\text{NS}}^{(s)} := \{u_0 \in H^s(\mathbb{T}^3) : \nabla \cdot u_0 = 0\}.$$

Define the (classical) blow-up locus

$$L_{\text{BlowUp}} := \left\{ u_0 \in M_{\text{NS}}^{(s)} : \exists T^* < \infty, \limsup_{t \uparrow T^*} \|\nabla u(t)\|_{L^\infty} = \infty \right\}.$$

Declared finite-dimensional search class. Exploration is restricted to a parametrized family $u_0(\theta)$ with

$$\theta \in \Theta \subset \mathbb{R}^d, \quad u_0(\theta) \in M_{\text{NS}}^{(s)},$$

generated by an explicit *search class generator* (e.g. vortex rings, Beltrami perturbations, multi-scale wave packets) together with explicit constraints ensuring divergence-freeness and energy bounds. All evidence is conditional on this declared class.

NS.2. Windows, definability, and measurement policy

Time horizon and windows. Fix a finite horizon $[0, T]$. Partition time into windows

$$W_j = [t_j, t_{j+1}), \quad 0 = t_0 < t_1 < \cdots < t_m = T,$$

where each W_j is given by a *definable formula* (o-minimal or Denef–Pas policy). This ensures finite enumeration of events and partitions in each bounded window.

Discretization and measurement policy (declared). A run is indexed by refinement parameters

$$\mathbf{r} = (\tau, \varepsilon, h, \Delta t, \{\alpha_\ell\}),$$

where τ is the truncation threshold, ε is optional smoothing scale, h is spatial grid scale, Δt is time step, and $\{\alpha_\ell\}$ is the filtration threshold set for persistence construction. All consequences are conditioned on the declared refinement schedule and test suite in NS.7.

NS.3. Realization into persistence (policy-bound)

Observable field. Let $\omega = \nabla \times u$ be vorticity, and define a scalar field

$$f_t(x) := |\omega(x, t)|.$$

Optionally apply smoothing $f_t^{(\varepsilon)} := \eta_\varepsilon * f_t$. Any smoothing sensitivity is charged to the ledger as measurement error.

Superlevel filtration and persistence object. Fix filtration thresholds $\{\alpha_\ell\}_{\ell=1}^L$ and define superlevel sets

$$X_{\alpha_\ell}(t) := \{x : f_t^{(\varepsilon)}(x) \geq \alpha_\ell\}.$$

A **declared** realization policy produces a constructible persistence object

$$F(t) = \mathfrak{P}_{\text{NS}}(u(t); \varepsilon, h, \{\alpha_\ell\}) \in \text{Pers}_k^{\text{cons}}.$$

All invariants used below are evaluated on $\mathbf{T}_\tau F(t)$, never on $F(t)$ directly.

After-collapse tower sampling. Choose sampling times $0 \leq s_0 < s_1 < \dots < s_N \leq T$. Consider the truncated tower on the B-side:

$$\mathbf{T}_\tau F(s_0) \rightarrow \mathbf{T}_\tau F(s_1) \rightarrow \dots \rightarrow \mathbf{T}_\tau F(s_N).$$

The tower policy (what is compared, how colimits/terminal maps are formed) is exactly the v17.0 canon.

NS.4. Diagnostics, Type IV rule, and defect potential (after-collapse)

Tower diagnostics (after \mathbf{T}_τ). Compute the v17.0 tower diagnostics $(\mu_{\text{NS}}, u_{\text{NS}})$ from the truncated tower. By canon:

$$\text{Normal} \iff (\mu_{\text{NS}}, u_{\text{NS}}) = (0, 0), \quad \text{Type IV} \iff (\mu_{\text{NS}}, u_{\text{NS}}) \text{ eq}(0, 0),$$

and the computation is performed strictly on $\mathbf{T}_\tau F(\cdot)$.

Ledger accounting in a quantale. Let \mathbf{V} be a declared commutative quantale with operation \oplus and order $\leq_{\mathbf{V}}$. Define ledger components (minimum set for NSE):

- δ^{meas} : measurement/smoothing/thresholding sensitivity,
- δ^{disc} : discretization error (grid/time step/tower sampling),
- δ^{alg} : algorithmic/implementation variance and finite arithmetic parts required by control tests,
- δ^{lift} : optional lifting cost (only if Lifter commits; otherwise $0_{\mathbf{V}}$).

Aggregate per state/window by

$$\Sigma \delta := \delta^{\text{meas}} \oplus \delta^{\text{disc}} \oplus \delta^{\text{alg}} \oplus \delta^{\text{lift}}.$$

Gap and admissibility (after-collapse). A declared policy provides an admissible margin gap_τ (possibly windowwise):

$$\Sigma\delta <_{\mathcal{V}} \text{gap}_\tau \implies \text{eligible for B-Gate}^+ \text{ attempt.}$$

The specific construction of gap_τ is a policy obligation (NS".3) and must be logged.

Defect potential (exploration score; never theorem-level). Define an exploration potential $\Phi_{\text{NS},\tau}$ on the after-collapse state, e.g.

$$\begin{aligned} \Phi_{\text{NS},\tau} := & w_\mu \mu_{\text{NS}} \\ & + w_u u_{\text{NS}} \\ & - w_\delta \|\Sigma\delta\|_{\mathcal{V}} \end{aligned} \tag{U.1}$$

with declared weights and declared reporting norm $\|\cdot\|_{\mathcal{V}}$. **No gate may depend solely on $\Phi_{\text{NS},\tau}$** ; it is used for navigation/prioritization only.

NS.5. Regimes and permitted outputs

Regime labeling (after-collapse). All regime labels use the B-side values:

- **Plain:** $(\mu_{\text{NS}}, u_{\text{NS}}) = (0, 0)$ and $\Sigma\delta <_{\mathcal{V}} \text{gap}_\tau$.
- **Noise:** $(\mu_{\text{NS}}, u_{\text{NS}}) = (0, 0)$ but margin fails: $\text{gap}_\tau \leq_{\mathcal{V}} \Sigma\delta$.
- **Singular (exploration label):** $(\mu_{\text{NS}}, u_{\text{NS}}) \text{ eq}(0, 0)$ and/or declared singular score exceeded.

Permissible outputs (only). This appendix allows the following outcomes:

- **Outcome A (negative evidence in class):** within declared Θ and declared refinement sweeps, no robust Type IV emergence beyond ledger inflation; all candidates fail robustness or anti-artifact tests.
- **Outcome B (audited candidate):** a robust Type IV candidate θ^* is produced with full reproducibility bundle, passing anti-artifact tests and remaining Type IV under refinement, *within the declared class*.
- **Outcome C (barrier/terminal):** lifting attempts and restarts fail within budgets; output a terminal barrier instance with complete artifacts (useful for B3-style disproof search, still [Spec]).

Appendix NS (continued): Refinement, Gates, and Reproducibility [Spec]

NS.6. Refinement sweeps and robust Type IV rule

Refinement schedule. A refinement sweep is a sequence $\{\mathbf{r}_m\}_{m \geq 0}$ with

$$h_{m+1} < h_m, \quad \Delta t_{m+1} < \Delta t_m, \quad \varepsilon_{m+1} \leq \varepsilon_m,$$

and either fixed τ or a declared τ -grid sweep (always gating after-collapse).

Robust Type IV candidate (strict rule). A candidate θ^* at fixed τ is *robust* only if there exists a refinement sweep \mathbf{r}_m such that:

- (R1) **Persistence of Type IV:** $(\mu_{\text{NS}}, u_{\text{NS}})(\theta^*; \mathbf{r}_m) \text{ eq}(0, 0)$ for all sufficiently large m , computed after \mathbf{T}_τ .

- (R2) **Ledger separation:** a declared normalized score separating defect growth from ledger inflation is non-decreasing (e.g. $\mu_{\text{NS}}/\|\Sigma\delta\|_{\text{V}}$, $u_{\text{NS}}/\|\Sigma\delta\|_{\text{V}}$, or a policy-defined alternative).
- (R3) **Anti-artifact tests pass:** NS.7 test suite is passed at each sufficiently refined level.
- (R4) **Replayability:** identical manifest + seed reproduces the candidate and its certificates.

NS.7. Anti-artifact test suite (mandatory)

T1: Filtration orientation sanity. Compare the superlevel filtration of f with an equivalent dual construction (e.g. sublevel of $-f$) under the same policy. Inconsistencies are charged to δ^{alg} and invalidate robustness.

T2: Smoothing sensitivity. Scan ε in a declared band $[\varepsilon_{\min}, \varepsilon_{\max}]$. If regime labels or Type IV status are not stable, charge δ^{meas} and invalidate robustness.

T3: Grid/time convergence. Run a convergence check across $(h, \Delta t)$ refinement. If diagnostics change qualitatively under refinement, charge δ^{disc} and invalidate robustness.

T4: Null-model controls. Require calibration on regimes expected to be benign within the interface: 2D NSE (global regular), 3D small-data regimes, and randomized low-vorticity controls. Failure indicates mis-calibration and invalidates the run.

NS.8. Gates and agent actions (NSE instance; after-collapse)

Gate Cascade (policy, after-collapse). For each window W and candidate θ , apply:

$$\text{B-Gate}^+ \rightarrow \text{PF/BC} \rightarrow \text{Overlap Gate}.$$

All inputs are computed on $\mathbf{T}_{\tau}F$. No pre-collapse value may enter.

B-Gate⁺ (eligibility). Attempt B-Gate⁺ only if:

$$(\mu_{\text{NS}}, u_{\text{NS}}) = (0, 0) \text{ and } \Sigma\delta <_{\text{V}} \text{gap}_{\tau}.$$

Otherwise record fail reason (margin fail or Type IV veto).

PF/BC (windowwise stability under policies). Check policy-functoriality / bar-compatibility *after-collapse* (as defined in v17.0). Any detected drift must be charged to $\delta^{\text{disc}} \oplus \delta^{\text{meas}}$ and re-tested.

Overlap Gate (Control \Rightarrow Overlap). Only after PF/BC passes, verify overlap consistency across adjacent windows and enforce arithmetic control only through overlap. Log the finite arithmetic part as δ^{alg} and record overlap certificates.

Restart and Summability. If repeated margin failures occur, invoke Restart Logic (windowwise) and enforce Summability under V . Stability bands (intervals of τ or refinement levels where decisions are constant) must be recorded.

Appendix NS': Toward a Soundness Layer (Obligations) [Spec]

Purpose. This section records theory-facing obligations required to make the NSE interface *sound enough* that exploration evidence meaningfully tracks analytic behavior. These are **obligations**, not achieved theorems.

NS'.1. Stability obligations for the realization

Audit Obligation W.1 (O1: Stable PDE-to-persistence realization). *Construct \mathfrak{P}_{NS} so that a declared stability bound holds, e.g.*

$$d_{\text{int}}(\mathfrak{P}(f), \mathfrak{P}(g)) \leq \|f - g\|_{L^\infty},$$

or an equivalent bound compatible with the v17.0 persistence metric and truncation policy.

Audit Obligation W.2 (O1b: Time regularity \Rightarrow tower sampling control). *If $\|f_t^{(\varepsilon)} - f_s^{(\varepsilon)}\|_{L^\infty} \leq L|t - s|$, prove that $t \mapsto \mathfrak{P}_{\text{NS}}(u(t))$ is Lipschitz in d_{int} , so sampling error is chargeable to δ^{disc} .*

NS'.2. Analytic-to-defect bridge targets (policy goals)

Dissipation-to-defect target. For a window $W = [t_0, t_1)$, set $\mathcal{E}_{\text{diss}}(W) = \int_{t_0}^{t_1} \|\omega\|_{L^2}^2 dt$. Target: bound after-collapse defect growth by

$$\mu_{\text{NS}} + u_{\text{NS}} \leq C(\tau) \cdot \mathcal{E}_{\text{diss}}(W) + (\text{explicit ledger terms}).$$

Total persistence vs analytic norms (target). Define $\text{TP}_p(F) := \sum_{b \in \text{Bar}(F)} \ell(b)^p$. Target: $\text{TP}_p(\mathfrak{P}(f)) \leq C(p) \|\nabla f\|_{L^p}^p$ for appropriate p under declared filtrations.

Truncation-compatible target. Define $\text{TP}_p^{(\tau)}(F) := \sum_{b \in \text{Bar}(\mathbf{T}_\tau F)} \ell(b)^p$. Target: $\text{TP}_p^{(\tau)}(\mathfrak{P}(f)) \leq C(\tau, p) \mathcal{N}(f)$ for a stable analytic quantity $\mathcal{N}(f)$.

Trimmed persistence \Rightarrow bounded tower defects (target). Target (policy): $\sup_t \text{TP}_p^{(\tau)}(F(t)) < \infty \Rightarrow \sup_t \mu_{\text{NS}}(t; \tau), \sup_t u_{\text{NS}}(t; \tau) < \infty$, modulo explicit ledger charges.

Appendix NS'': Proof-first Program (Obligations and Loop) [Spec]

NS''.1. Core obligations (if one aims beyond exploration)

Audit Obligation X.1 (O2: Blow-up completeness within the interface). *If a strong solution loses regularity at T^\star , show that for some declared (τ, ε) , $\mu_{\text{NS}}(t; \tau)$ or $u_{\text{NS}}(t; \tau)$ diverges as $t \uparrow T^\star$, beyond ledger inflation.*

Audit Obligation X.2 (O3: Analytic a priori control \Rightarrow bounded AK defects). *Using NSE inequalities and bridge targets in NS', prove uniform bounds on μ_{NS} , u_{NS} for fixed τ , and quantify all remaining uncertainty as ledger terms.*

Audit Obligation X.3 (O4: Close the loop). *Combine O2 and O3 to rule out Type IV within the interface, thereby excluding blow-up inside the declared framework. Any such result, if achieved, must be stated as a theorem in the main text, not in this appendix.*

NS".2. Calibration oracle checkpoint (mandatory for credibility)

Require the pipeline to remain **Plain** on 2D NSE and on 3D small-data regimes within ledger budgets. Treat failures as invalidation of the exploration configuration.

NS".3. Gap estimation obligation (policy completeness)

Audit Obligation X.4 (O5: Construct gap_τ). *Provide a computable policy for $\text{gap}_\tau = \text{gap}(\tau; T, u, \text{policy})$ from logged quantities, controlling admissible ledger inflation for gate eligibility.*

NS Run Artifacts: Reproducibility Bundle [Spec]

Required directory layout.

```
runs/  
  NS-YYYYMMDD-HHMMSSZ_<shortid>/  
    run.yaml  
    ledger.json  
    diag.csv  
    proof.log  
    artifacts/  
      barcodes/  
      diagrams/  
      fields/  
      checkpoints/
```

Artifact A: run.yaml (template; canon-aligned)

```
meta:  
  run_id: "NS-YYYYMMDD-HHMMSSZ_ab12cd"  
  created_utc: "YYYY-MM-DDTHH:MM:SSZ"  
  engine: {name: "AK-HDPST", version: "v17.0", mode: "Exploration", spec_level: "[Spec]"}  
  canon:  
    after_collapse_only: true  
    typeIV_rule: "(mu,nu)!=(0,0) after T_tau"  
    gate_order: ["B_GatePlus","PF_BC","OverlapGate"]  
    proxy_separation: true  
    quantale_ledger: true  
  
problem:  
  equation: "3D incompressible NSE"  
  domain: "T3"  
  viscosity_nu: 0.001  
  horizon: {t_start: 0.0, t_end: 2.0}  
  function_space: {s: 3.0, divergence_free: true}  
  
windows:  
  base: ["W1","W2","W3","W4"]  
  definable_formulas:  
    - "W1: 0.0 <= t < 0.5"  
    - "W2: 0.5 <= t < 1.0"
```

```

- "W3: 1.0 <= t < 1.5"
- "W4: 1.5 <= t < 2.0"

search_class:
  theta_dim: 18
  generator:
    name: "vortex_rings_collide"
    constraints: {enforce_div_free: true, energy_bound: true}

pde_solver:
  method: "pseudo_spectral"
  grid: {n: 256, aliasing: "2/3"}
  time_integration: {integrator: "RK3", dt: 1.0e-4}

observable:
  scalar_field: {definition: "|omega|"}
  smoothing: {enabled: true, eps: 0.015, ledger_tag: "delta_meas"}

persistence:
  category: "Pers_k^cons"
  filtration: {type: "superlevel", field: "f_t_eps"}
  thresholds: {count: 64}
  homology_degrees: [1,2]

truncation:
  T_tau: {enabled: true, tau: 0.02, policy: "fixed"}

tower:
  sampling: {n_times: 41}
  indicators:
    mu: {definition_policy: "v17_colim_to_terminal_cokernel", degree: 1}
    nu: {definition_policy: "v17_kernel_or_cokernel_variant", degree: 1}

ledger_policy:
  quantale: {name: "V_addxV_max", op: "product", order: "coordinatewise", report_norm: "L1"}
  components: ["delta_meas", "delta_disc", "delta_alg", "delta_lift"]

gates:
  B_GatePlus:
    require: ["after_collapse_only", "delta_total < gap_tau", "(mu,nu)=(0,0)"]
  PF_BC:
    enforce: true
  OverlapGate:
    enforce: true
    control_implies_overlap: true

tests:
  anti_artifact_suite:
    T1_filtration_orientation: true
    T2_smoothing_sensitivity: true
    T3_grid_dt_convergence: true
    T4_null_model_controls: true

```

Listing 6: NS Exploration Run Template: run.yaml

Artifact B: ledger.json (minimal schema; quantale)

```
{
  "run_id": "NS-YYYYMMDD-HHMMSSZ_ab12cd",
  "quantale_policy": { "name": "V_addxV_max", "op": "product", "order": "coordinatewise", "report_norm": "L1" },
  "delta_components": {
    "delta_meas": { "value": 1.20, "confidence": 0.7 },
    "delta_disc": { "value": 2.10, "confidence": 0.6 },
    "delta_alg": { "value": 0.80, "confidence": 0.8 },
    "delta_lift": { "value": 0.00, "confidence": 1.0 }
  },
  "delta_total": { "value": 4.10 }
}
```

Listing 7: Ledger Template: ledger.json

Artifact C: diag.csv (schema; after-collapse only)

```
run_id,theta_id,t>window,tau,eps,grid_n,dt,thresholds_count,mu,nu,phi_ns,delta_meas,delta_disc,
delta_alg,delta_lift,delta_total,flags
```

Listing 8: Diagnostics Time Series: diag.csv (schema)

Artifact D: proof.log (JSONL; anti-artifact and gate replay)

```
{ "run_id": "...", "stage": "B_GatePlus", "status": "PASS", "details": "..." }
{ "run_id": "...", "test": "T1_filtration_orientation", "status": "PASS", "details": "..." }
{ "run_id": "...", "test": "T3_grid_dt_convergence", "status": "PASS", "details": "..." }
{ "run_id": "...", "stage": "OverlapGate", "status": "PASS", "details": "..." }
```

Listing 9: Test and Gate Output: proof.log (JSONL)

Validity requirement. Any Type IV claim (even as [Spec] evidence) is invalid without run.yaml, ledger.json, diag.csv, proof.log, and the referenced persisted artifacts sufficient for deterministic replay.

Concluding Remarks and Acknowledgments (v17.0, AK–HDPST + HDPS)

Standing scope, coefficients, windows, and UCC guard–rails. Unless stated otherwise, coefficients lie in a *field* k . All *Core* statements live in constructible one-parameter persistence over a field and, when realized, target $D^b(k\text{-mod})$. In [Spec] appendices that use sheaf-theoretic, arithmetic, or Fukaya-categorical realizations we write the coefficient field as Λ ; this is only a notational change and does not alter the *Core* category or its hypotheses.

Filtered (co)limits are computed *objectwise* in $[\mathbb{R}, \text{Vect}_\Lambda]$ under the scope policy (Appendix A) and returned to the constructible subcategory by verification or by applying \mathbf{T}_τ at the persistence layer. All statements are made under the *Unified Collapse Contract (UCC)* (Thm. 1.1, Chapter 1): we work in constructible 1D persistence, with

- a commutative Quantale $(V, \oplus, \leq, 0)$ enriching the time index and hosting all error budgets;

- *right-open*, MECE windows along a τ -sweep, which, when required, are *Denef–Pas definable* to guarantee finite event sets and finite Čech depth (Appendix Q);
- an *after-collapse policy*: all equalities, exactness claims, monotonicity statements, comparisons, and gluing are asserted only after applying \mathbf{T}_τ at the persistence layer.

Within these guard-rails, the δ -ledger aggregates all residuals in V , is subadditive under composition, and is non-increasing under after-collapse 1-Lipschitz post-processing; this allows $\Sigma\delta$ to be used both as an audit budget and, in **[Spec]** mode, as a scalar *Defect Potential* Φ_τ for high-dimensional search (Ch. 13).

Canon commitments (v17.0). The following conventions are fixed throughout and are not overridden by any application or search layer:

- **After-collapse semantics.** Any assertion that can affect a proof object (gate pass/fail, regime classification, certificate export) is evaluated on the B-side $\mathbf{T}_\tau(\mathbf{P}_i(F))$ at the persistence layer. Pre-collapse readings may be recorded as diagnostics only, never as proof-bearing evidence.
- **Type IV meaning (fixed).** “Normal” means $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$ and **Type IV (invisible obstruction)** means $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) \text{ eq}(0, 0)$, computed *after* \mathbf{T}_τ under the declared tower policy.
- **No global equivalence.** We never assert a global equivalence $\text{PH}_1 \iff \text{Ext}^1$. Only the one-way bridge $\text{PH}_1 \Rightarrow \text{Ext}^1$ is Core, and any reverse direction is **[Spec]** and allowed only under explicit safety programs and tests.
- **Core vs [Spec] boundaries are auditable.** Any external realization (sheaf/arithmetic/Fukaya/spectral proxies) is either proved in the Core target $D^b(k\text{-mod})$ or labeled **[Spec]** with explicit δ -budgeting and gate restrictions.

What is proved (Core; F0–F6 & P1–P10, UCC/bridge extensions). Within the above regime we establish machine-checkable results and arrange them for formalization.

- *Exact truncation and filtered lift (F1–F2).* The Serre reflector \mathbf{T}_τ deletes precisely bars of length $\leq \tau$, is exact, idempotent, and 1-Lipschitz (indeed V-1-Lipschitz under Quantale enrichment). It admits a filtered lift C_τ unique up to f.q.i. with $\mathbf{P}_i(C_\tau F) \cong \mathbf{T}_\tau(\mathbf{P}_i F)$ (Appendices A/B).
- *CNF and field edge (P1–P2).* After collapse, objects split in $D^b(k\text{-mod})$: $X \simeq \bigoplus_i H^i(X)[-i]$, and $\text{Ext}^1(X, k) \cong \text{Hom}(H^1(X), k)$.
- *Bridge (one-way) and scope of reverse.* Under a t -exact realization of amplitude ≤ 1 , $\text{PH}_1(F) = 0 \Rightarrow \text{Ext}^1(\mathcal{R}(F), k) = 0$ (Appendix C). Any reverse implication remains **[Spec]** and is admitted only through explicit safety programs with independent quantitative separation tests (Chapter 16; Appendix W).
- *Safe low-pass (P4).* Even, mass-1, bandwidth $\propto \sqrt{\tau}$ kernels commute with \mathbf{T}_τ up to a controlled δ^{alg} and keep $\mathbf{T}_\tau \circ L_\tau$ 1-Lipschitz (Prop. 2.10; Ch. 2/Appendix E).
- *Monotonicity vs. stability (P5).* Deletion-type updates are non-increasing after collapse; inclusion-type updates are non-expansive.
- *Convergence Manager (P6).* For countable *Denef–Pas* covers of finite Čech depth, the Quantale-summed error satisfies $\sum \delta < \infty$ and overlap gluing holds globally (Thm. 1.10; Appendices J/Q).

- *AWFS 2-cell additivity (P7)*. 2-cell defects add subadditively in the Quantale (Ch. 5; Appendices K/L).
- *Gate calculus with cut elimination (P8)*. The default cascade is operated as a sequent calculus with fixed ordering:

$$E_1=0 \implies (\mu, u)=(0,0) \implies \text{Ext}^1=0 \implies \text{PH}_1=0.$$

Success later never overturns failure earlier.

- *Stability bands (P9)*. Open τ -intervals with $(\mu, u)=(0,0)$ certify stability; Type-IV is excluded in conjunction with P6 (Ch. 4; Appendix D/J).
- *Reproducibility theorem (P10)*. From pass-logs of T-ExtZero-implies-PHZero, T-Countable-Cover, T-Delta-Sum-Converges, T-Lipschitz-AfterCollapse, T-Exactness-Persistence (and arithmetic T-Iwasawa-Alignment), the P3/P6/P8 conclusions are mechanically reconstructed.
- *UCC collapse nucleus and Quantale ledger (extension)*. The Unified Collapse Contract (Thm. 1.1) upgrades \mathbf{T}_τ to a V-nucleus and shows that the Quantale-valued δ -ledger is subadditive, non-increasing under after-collapse post-processing, and therefore sound as a global potential $\Sigma\delta$ for both audit and (under [Spec] policies) navigation (Ch. 1/Appendix S).

What is specified and how it is audited ([Spec]). All [Spec] items are explicitly contracted to be *non-expansive after truncation* and are audited by the windowed diagnostics (μ, u) together with a Quantale-valued δ -ledger. In particular, **Type IV is always a post- \mathbf{T}_τ phenomenon** and never “many short bars” (short bars are deleted by design).

- *UCC search layer and dual mode*. The Quantale enrichment $(V, \oplus, \leq, 0)$, definable windows (Denef–Pas preferred), and the AWFS view $\text{Id} \Rightarrow L \dashv R \Rightarrow \text{Id}$ with $R = C_\tau$ (Ch. 1/5; Appendices K/L) provide the *Audit Mode* in which $\Sigma\delta < \text{gap}_\tau$ certifies validity. In *Navigation Mode* (Part II), the same $\Sigma\delta$ is scalarized into a Defect Potential Φ_τ (with Type-IV penalties) used by AI agents under Chapter 13, but never to override the audit gates.
- *Mirror/Transfer pipelines*. A natural 2-cell $\text{Mirror} \circ C_\tau \Rightarrow C_\tau \circ \text{Mirror}$ with uniform bounds $\delta(i, \tau)$ yields δ -controlled commutation; bounds add along pipelines and are non-increasing under 1-Lipschitz post-processing (Appendix L).
- *Multi-axis reflectors*. For exact reflectors from hereditary Serre subcategories, nesting is order-independent; otherwise an A/B test with tolerance η and deterministic fallback is used. Residuals Δ_{comm} are recorded as δ^{alg} (Appendix K).
- *Arithmetic layers (SCTF/ECF/Iwasawa)*. Local traces (Igusa/Tate) couple to post-collapse measurements; discrepancies externalize to $\delta_{\text{alg}}, \delta_{\text{meas}}$. The *Explicit-Contract Formula* (ECF) enforces the safety-side inequality

$$|\mu_{\text{Coll}}(W, \tau) - \langle \text{Obs}(R_{\text{spec}}(F), W), \varphi_\tau \rangle| \leq \varepsilon_{\text{tot}}(W, \tau),$$

with RHS fully represented in the δ -ledger (Appendix M). The *Iwasawa Gate* aligns $(\mu_{\text{Collapse}}, \mu_{\text{Iwasawa}})$ in a three-state regime (lower bound / match / drift-corrected) to suppress Type-IV drift (Ch. 7; Appendix R). Here μ_{Collapse} and the classical Iwasawa μ are distinct invariants and are never identified.

- *Fukaya realizations*. Action filtration yields constructible persistence on bounded windows; continuation is 1-Lipschitz and adding stops is deletion-type (Appendix O).

- *PF/BC transport*. Projection formula/base change are transported only through the after-collapse protocol; residual discretization/measurement slack is budgeted as $\delta^{\text{disc}}, \delta^{\text{meas}}$ (Appendix N).
- *HDPS engine, Validity Map, and AI agents*. Part II and Appendices U/V/W specify the *High-Dimensional Projection Search (HDPS)* layer:
 - the Defect Potential Φ_τ and its stratification into Valid/Noise/Sing regimes and a terrain-cell decomposition (Ch. 13; Appendix V);
 - the autonomous agents Hunter/Mapper/Lifter with formally defined operational semantics and the Hunter Action Log schema (Ch. 14–15; Appendix U);
 - the *Validity Map* and *Global Certificate* as verifiable graph structures (Appendix V);
 - the Bridge Programs B1/B2/B3 and the Spectral-Gap Condition, which govern when a local reverse implication $\text{Ext}^1 \Rightarrow \text{PH}_1$ is permitted under **[Spec]** guard-rails (Ch. 16; Appendix W).

All such search-side components are constrained by the UCC budget: they may propose paths and lifts, but the Core vetoes any step with $\Sigma \delta \geq \text{gap}_\tau$.

- *Case studies (templates only)*. Application templates—e.g. the Navier–Stokes case study (Appendix NS)—translate classical questions into the AK-HDPST language of truncation, tower diagnostics, Type IV veto, and reproducible evidence. They remain **[Spec]** and do not alter the Core theorems.

Operational pipeline (end-to-end, canon order). Per window and degree: enforce B–Gate⁺ with safety margin $\Sigma \delta < \text{gap}_\tau$ and the **Type IV veto** $(\mu, u) = (0, 0)$ (both evaluated after \mathbf{T}_τ); then apply the post-collapse *PF/BC* checks; and finally apply the *Overlap Gate* (post-collapse equivalence up to budget, Čech control, and alignment constraints). Across windows, Restart/Summability (κ -restart and $\Sigma \delta < \infty$) paste local certificates into global ones (Appendix J), yielding a verifiable global object (Appendix V). A single Quantale-sum \oplus aggregates pipeline budgets: Mirror–Collapse bounds $\delta(i, \tau)$, A/B residuals Δ_{comm} , discretization and measurement terms, and, in HDPS mode, lifting penalties δ^{lift} .

Reproducibility, formalization, and tests (run.yaml v17.0). Appendix G specifies the manifest run.yaml (Quantale, definable window formulae, AWFS/2-cell bounds, τ -sweeps, spectral and lifting policy, search strategy) and artifact schemas with canonical serialization and cross-linked hashes. Appendix F outlines a Lean/Coq pathway for Core components (Serre localization; 1-Lipschitz; comparison maps and (μ, u) ; CNF and field edge; the one-way bridge; and API stubs for budgets and controlled commutation). Appendix U prescribes the Hunter Action Log and Proposer/Verifier split; Appendix V formalizes the Global Certificate; Appendix W the Bridge Programs. Chapter 12 provides tests for V-Lipschitz laws, definable coverage/finite Čech depth, deletion-type monotonicity, filtered-colimit behavior, Mirror/tropical pipelines, A/B soft-commuting, Restart/Summability, arithmetic alignment (T-Iwasawa-Alignment), and—in HDPS mode—stability of Φ_τ under resolution changes. Optional tests include T-PFBC-AfterCollapse and T- \mathcal{N}_{len} .

Limitations and guard-rails. All claims are confined to the implementable persistence/spectral/categorical layers *after collapse*. No claim of a global equivalence $\text{PH}_1 \Leftrightarrow \text{Ext}^1$ is made; only the one-way implication under (B1)–(B3) and a locally certified reverse under the Spectral-Gap Condition are used, and then only within definable windows and explicit budgets. Spectral indicators are not f.q.i. invariants; they are evaluated under fixed policies with deletion-type monotonicity and general non-expansiveness. The collapse diagnostic μ_{Collapse} is distinct from the classical Iwasawa μ . The HDPS/AI layer (Hunter/Mapper/Lifter, Bridge

Programs, Validity Maps, case studies) is entirely **[Spec]**: it can *propose* search trajectories and certificates, but only the Core/UCC can *certify* them.

Outlook. Future work will refine quantitative links between persistence energies and spectral tails, broaden verifiable criteria for stability bands and $(\mu_{\text{Collapse}}, u_{\text{Collapse}}) = (0, 0)$, and extend formal libraries (shift/interleaving; PF/BC transport; budget calculus) together with domain templates (arithmetic/Langlands/PDE/-Fukaya) equipped with auditable δ -controls. On the HDPS side, further development of Bridge Programs, Validity Maps, and domain-specific case studies (NSE, BSD, RH, Langlands) may clarify when AI-assisted exploration can be safely promoted to Core-level proofs.

References

- [1] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*, American Mathematical Society, 2010.
- [2] F. Chazal, V. de Silva, M. Glisse, and S. Oudot, *The Structure and Stability of Persistence Modules*, SpringerBriefs in Mathematics, Springer, 2016.
- [3] M. Lesnick, The Theory of the Interleaving Distance on Multidimensional Persistence Modules, *Foundations of Computational Mathematics* **15** (2015), no. 3, 613–650.
- [4] C. A. Weibel, *An Introduction to Homological Algebra*, Cambridge University Press, 1994.
- [5] P. Gabriel and M. Zisman, *Calculus of Fractions and Homotopy Theory*, Springer, 1967.
- [6] L. van den Dries, *Tame Topology and o-minimal Structures*, Cambridge University Press, 1998.
- [7] K. I. Rosenthal, *Quantales and Their Applications*, Pitman Research Notes in Mathematics Series 234, Longman, 1990.
- [8] R. Garner, Understanding the Small Object Argument, *Applied Categorical Structures* **17** (2009), no. 3, 247–285.

Acknowledgments. The manuscript was prepared by the author, with assistance from an AI tool (ChatGPT) through an iterative workflow; any remaining errors are the author’s responsibility.

Final note. The separation between the provable *Core* and auditable **[Spec]** contracts, together with the after-collapse order, Quantale-aggregated δ -budgets, Restart/Summability, and the HDPS engine, provides a conservative and extensible methodology for cross-domain reuse within the guard-rails of **v17.0 (AK-HDPST + HDPS)**.