

A Communication Theory of Computation Realization

Resource-Conditioned Geometry of Legal Realizations and the Safe-Commitment Law

Fitih M. Cinnor for semiAIfoundry Research

Research Lead and Accountable Author

semiAIfoundry Research, a research group of Semi AI Foundry LLC

Accountable correspondence: fcinnor@semiaifoundry.com

22 August 2026

Version 3.2.0

DOI: [10.5281/zenodo.22048090](https://doi.org/10.5281/zenodo.22048090) Version: 3.2.0

Executable companion: <https://github.com/semiaifoundry/computation-realization-geometry>

Collective scholarly attribution and research provenance. This work was produced by semiAIfoundry Research through a human-directed collaboration across the intelligence continuum—human, artificial, and hybrid. AI models and multi-agent systems made material contributions to mathematical exploration, formal derivation, computational implementation, verification, literature analysis, adversarial critique, and manuscript development. Fitih M. Cinnor led the research program and serves as the accountable human author, retaining responsibility for the scientific claims and corrections.

Abstract

A computation specifies a semantic relation, not one inevitable algorithm, hierarchy, architecture, or physical implementation. This article introduces *Computation–Realization Geometry* (CRG), a computation-first theory of the legal ways a declared computation can become physical. A computation and resource contract define a hierarchy-indexed realization family $\mathcal{P}$; typed signature maps produce the exact attainable region $\mathcal{R}$, the upper obligation region Γ visible to continuous monotone resource decisions, and the convex upper body $\mathcal{K}$ visible to additive nonnegative prices. The Safe-Commitment Law gives necessary and sufficient conditions for a retained realization family to preserve every decision in either declared class. A strict engineering-cost separation corollary constructs a continuous, coordinatewise strictly increasing, coercive regret witness whenever the required geometry is lost.

The theory assembles contextual-state, hierarchy, and circuit–protocol constructions from established residual-state and communication-complexity arguments; establishes an exact interactive directional cut-rank orthant for finite-field linear computations; identifies deterministic carrier realizations with zero-error network function computation under matched contracts; and adapts convex support duality, temporal cut-state conservation, regular variation, positive-system contraction, and semialgebraic phase analysis to typed computation realizations. It reclassifies Rent-like laws as outputs of a declared realization, hierarchy, accounting convention, estimator, and fitting window. Technology is applied after legality and attainability, yielding a formal distinction between repricing an unchanged realization family and reopening it when capability or legality changes.

The framework is carried through matrix multiplication, exact attention, layout conversion, a complete evaluation within a declared finite Transformer registry and typed exclusions, a graph-to-certificate reference representation, a controlled retention experiment, and compact routed 2.5D/3D realizations. The work supplies a unified abstraction layer linking semantic obligation, legal realization, decision-sufficient retention, scaling, architecture selection, and physical evidence.

Reader map

Reader question	Recommended path
What is CRG and why is it needed?	Introduction, abstraction ladder, and Safe-Commitment Law.
What are the semantic and exact foundations?	Contextual states, circuit-protocol translations, interactive linear orthants, and conservation.
How does the scaling theory work?	Fixed-realization profiles, regular variation, hierarchy phase, positive modes, curvature, and caps.
How does CRG guide physical design?	Technology phases, certified brackets, pruning, scale matching, link laws, and component certificates.
What does an end-to-end case look like?	Transformer registry, retention experiment, and routed 2.5D/3D study.
What is executable?	The public CRG repository, schemas, validation index, retained baselines, and release manifest.

Contents

1	Introduction	6
1.1	Three layers of the theory	6
1.2	The abstraction ladder	7
1.3	The Safe-Commitment Law	8
1.4	Theory architecture and principal results	8
1.5	Mathematical domain and extension space	8
2	Historical progression and relation to prior work	9
2.1	From computability to realization	9
2.2	Information, communication, and decision sufficiency	9
2.3	Communication complexity, distributed function computation, and network coding	10
2.4	VLSI area-time, I/O complexity, and parallel-machine models	10
2.5	Rent's rule, graph structure, and multiscale measurement	10
2.6	Platforms, interfaces, contracts, and design rules	10
2.7	Set-based design, tradespaces, and systems architecture	11
2.8	Program transformations, intermediate representations, and equality saturation	11
2.9	Vector optimization, Pareto sets, and robust decisions	11
2.10	Architecture exploration, compiler mapping, and physical EDA	12
2.11	Physical realization, reversibility, and thermodynamic cost	12
2.12	Comparative synthesis	12
2.13	The gap closed by computation-realization geometry	13
I	Semantic contracts and exact resource geometry	13
3	Terminalized computation and resource contracts	13
3.1	How a resource contract changes the realization family	14
4	Canonical distributed interface states	15
5	General causal realizations and exact certificates	17
5.1	Finite causal protocols and circuit translations	17
5.2	Where interaction changes the signature	19
5.3	Complete interactive finite-field linear orthant	19
5.4	Carrier graphs and network computation	21
5.5	Temporal cut-state conservation and scoped projections	22

6	The CRG abstraction ladder and decision-sufficient geometry	23
6.1	Why the geometry has four levels	24
6.2	Invariance and coherent hierarchy maps	25
6.3	Monotone reconstruction and linear duality	26
6.4	Safe commitment and insufficiency of scalar summaries	27
7	Monotonicity and composition	29
II	Scaling laws from resource geometry	31
8	A scale profile from one legal realization	31
9	Stable dilation limits and finite hierarchy phase	32
9.1	Preferred dilation factors	32
9.2	Normalized dilation action and phase ratios	33
10	Windowed estimators and finite-scale bias	34
11	Recursive mechanisms and certified scaling basins	35
12	Arithmetic modes and finite-scale curvature	38
13	Interface reach and terminal-boundary crossover	40
14	Finite-scale observability and failure taxonomy	42
14.1	Deterministic mechanism checks	44
15	Public-netlist diagnostics and mode-resolved attribution	44
15.1	Registered measurement and reproduction protocol	44
15.2	Hierarchy- and estimator-conditioned scalar fits	47
15.3	Exact mode identity, finite-scale failure, and transfer	48
15.4	Interpretation of the diagnostic	49
III	Technology-conditioned realization and certified design	51
16	From attainable geometry to physical decisions	51
16.1	Pareto sufficiency and price-conditioned regimes	52
16.2	Certified inner and outer descriptions	53
16.3	Lossless retention for a technology domain	53
17	Architecture–technology scale and link matching	54
17.1	Scale-dependent communication strain	54
17.2	Aligned links and chiplet interfaces	54
18	Reference intermediate representation and compiler boundary	55
19	Analytic component certificates	56
19.1	Dense matrix multiplication across memory levels	56
19.2	A certified 2.5D technology map	57
19.3	Exact attention: recomputation versus storage	58
19.4	Cross-stage state and layout laws	59
IV	End-to-end certificate	61

20 Certified nonseparable co-design of a long-context Transformer superblock	61
20.1 Workload and execution contract	61
20.2 Evidence-tiered technology contract	62
20.3 Finite architecture and mode registry	62
20.4 Conserved timing and energy	63
20.5 Strict registered decisions and phase walls	64
20.6 Context scaling and twelve-contract sensitivity	65
20.7 Evidence ledger and falsification conditions	66
V Executable realization and end-to-end validation	67
21 From a compiler graph to physical consequence	67
21.1 Reference CRG intermediate representation	67
21.2 Head-to-head retention experiment	68
21.3 Carrying one semantic obligation into routed 2.5D and 3D models	69
21.4 Integrated computation-to-technology statement	70
VI Synthesis and extension	70
22 A unified evidence architecture	70
23 The scientific object established here	71
24 Extension domains	72
24.1 Stochastic, approximate, and reversible realizations	72
24.2 Automatic realization-family construction	72
24.3 Physical and manufacturing geometry	72
24.4 Dynamic lifecycle geometry	72
25 Implications for computation-to-technology design	72
26 Conclusion	73
VII Appendices	74
A Harmonized notation and terminology	74
B Contract precedence and quantifier order	75
C Essential reusable contract clauses	75
C.1 Interaction, traffic, and time	75
C.2 Layout conversion	75
C.3 Scaling and boundary modes	75
C.4 Finite-registry design decisions	75
C.5 Transformer certificate	76
D Worked derivations of the CRG abstraction ladder	76
D.1 Broadcast and parity: semantic state versus physical incidence	76
D.2 INDEX: interaction changes the attainable geometry	76
D.3 Exact points, upper obligations, and price-visible bodies	77
D.4 Safe commitment and explicit regret	77
D.5 From a compiler graph to CRG objects	78
D.6 Repricing versus reopening	78
D.7 Why monotone observations reconstruct the upper region	78
D.8 Why additive prices reconstruct the convex upper body	79

D.9 A two-architecture phase wall	79
D.10 Deriving the architecture–technology scaling law	79
D.11 Transformer bridge arithmetic from ownership semantics	80
D.12 Deriving retention regret in the reference experiment	80
E Derivation map from theorem to engineering artifact	80
Collective attribution, accountability, and AI research provenance	87

1 Introduction

A computation is not yet a machine. It specifies a relation among inputs, outputs, and admissible error; it does not by itself choose an algorithm, interaction pattern, schedule, data layout, memory policy, hierarchy, architecture, or physical substrate. A physical computer must make all of those choices. It transforms state locally, preserves selected distinctions through time, and makes other distinctions available nonlocally across wires, memories, chiplet interfaces, package fabrics, active tiers, boards, and networks.

The passage from computation to physical system therefore contains a family of legal realizations rather than one inevitable graph. A result may be stored or recomputed; a function may be replicated or shared; two stages may be fused or separated; a protocol may be one-way or interactive; data may be serialized, partitioned, coded, approximated, or moved to another hierarchy level. Each choice changes several obligations at once. Lower movement may require more state. Lower state may require more work. Higher replication may reduce traffic while increasing local capacity and synchronization. Specialization may reduce arithmetic while creating a new ownership boundary. A new package or vertical capability may make a previously unattractive realization preferable.

Classical theories expose important parts of this problem. Communication complexity establishes information that must cross a declared cut. I/O and area-time theories establish service limits in declared memory and VLSI models. Network coding enlarges the set of legal transformations on a fixed carrier. Rent analysis describes multiscale connectivity in a realized graph under a chosen hierarchy and estimator. Compiler and architecture systems construct and search large spaces of schedules, mappings, and hardware instances. Physical design and DTCO/STCO flows evaluate realizations against specific technologies. The unresolved object lies one level earlier: the computation-conditioned family from which those different projections are taken.

This paper develops *A Communication Theory of Computation Realization*. Its central mathematical object is *computation-realization geometry* (CRG): the resource-conditioned geometry of the legal ways a declared computation can become physical, the jointly attainable typed obligations those realizations impose, and the exact information retained by progressively coarser decision classes. Its foundational question is

Which distinctions must become available across which boundaries, which legal realizations can provide them jointly, and what information must a design representation preserve until technology and objectives select among them?

The word *communication* is used in its semantic and physical senses. Semantically, it is the process by which distinctions required on one side of a boundary become available there when they are not already retained as state. Physically, those distinctions are encoded in messages, nets, memory transfers, collectives, links, and routed carriers. CRG keeps the two levels connected without identifying them: a semantic obligation constrains every in-contract realization, while a technology determines how that obligation is encoded, served, and priced.

1.1 Three layers of the theory

The theory separates three objects that are frequently collapsed.

Computation.

A terminalized map F specifies required outputs, terminal ownership, and correctness. It answers *what must be accomplished*.

Realization contract.

A resource contract ρ and hierarchy $\mathcal{T}$ specify legal local primitives, interaction, work, depth, storage, replication, recomputation, approximation, accounting, and hierarchy rules. They answer *how the computation may legally be realized*.

Technology.

A technology contract τ specifies physical capabilities, embeddings, service laws, and costs. It answers *which legal realization is feasible and preferred in a physical world*.

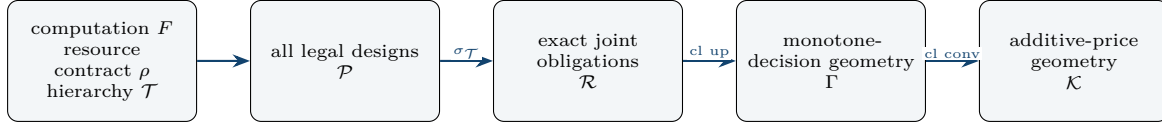
The order is load bearing:

$$\boxed{\text{computation and resource contract} \longrightarrow \text{legal realization geometry} \longrightarrow \text{technology-conditioned selection.}} \quad (1.1)$$

A price change can expose a different point in an unchanged legal geometry. A capability change can admit new primitives, schedules, storage modes, or hierarchies and thereby changes the legal family itself. The former is a repricing operation; the latter is a change of CRG slice.

1.2 The abstraction ladder

Fix a terminalized computation F , a resource and accounting contract ρ , and an admitted hierarchy $\mathcal{T}$. Let $\mathcal{P}_F^\rho(\mathcal{T})$ denote the legal realization family. A typed signature map $\sigma_{\mathcal{T}}$ sends each realization to the jointly provisioned quantities that the declared observation retains: interface state, event count, traffic, work, causal depth, activity, replication, distributions, or other typed coordinates.



Moving right increases abstraction. Each step is lossless exactly for the future decision class written inside the destination box. Technology is applied after the legal family and the required observation object have been defined.

Figure 1: The computation-realization geometry abstraction ladder. The theory separates realization identity, exact joint attainability, monotone-obligation information, and additive-price information rather than treating them as one interchangeable design summary.

The central CRG representation is

$$(F, \rho, \mathcal{T}) \implies \mathcal{P}_F^\rho(\mathcal{T}) \xrightarrow{\sigma_{\mathcal{T}}} \mathcal{R}_F^\rho(\mathcal{T}) \longrightarrow \Gamma_F^\rho(\mathcal{T}) \longrightarrow \mathcal{K}_F^\rho(\mathcal{T}). \quad (1.2)$$

The four retained objects have distinct meanings.

Table 1: The CRG abstraction ladder in words and symbols.

Retained information	Symbol	Questions preserved
All legal realizations	$\mathcal{P}$	Architecture identity, event structure, schedule, causal organization, and embedding-sensitive questions
Exact jointly attainable obligations	$\mathcal{R}$	Which discrete resource combinations are realized by at least one legal design
Closed upper obligation geometry	Γ	Every continuous coordinatewise-monotone resource objective
Convex price-visible body	$\mathcal{K}$	Every additive nonnegative-price objective and explicitly admitted amortized mixture

The exact image

$$\mathcal{R}_F^\rho(\mathcal{T}) = \sigma_{\mathcal{T}}(\mathcal{P}_F^\rho(\mathcal{T}))$$

retains joint attainability but forgets which realizations share one signature. The upper obligation region

$$\Gamma_F^\rho(\mathcal{T}) = \text{cl}(\text{up } \mathcal{R}_F^\rho(\mathcal{T}))$$

adds provisioned points that consume at least as much of every retained resource. This operation loses no information for continuous monotone costs, because increasing a resource coordinate cannot improve such a cost. The convex upper body

$$\mathcal{K}_F^\rho(\mathcal{T}) = \text{cl conv } \Gamma_F^\rho(\mathcal{T})$$

forgets nonconvex distinctions that no additive nonnegative price can observe. It is therefore the exact price quotient, but not the exact quotient for every nonlinear or realization-sensitive objective.

A two-resource example makes the information loss concrete. Suppose one legal realization has signature $(0, 2)$ and another has $(2, 0)$, where the coordinates are movement and fast state. The coordinatewise minima suggest $(0, 0)$, but no legal realization attains it. The convex body contains $(1, 1)$, which is sufficient for linear-price interpolation but can give the wrong answer to the nonlinear bottleneck objective $\max\{x_1, x_2\}$. Thus no one of $\mathcal{P}, \mathcal{R}, \Gamma, \mathcal{K}$ can be substituted for all the others.

1.3 The Safe-Commitment Law

Let $\emptyset \neq \mathcal{A} \subseteq \mathcal{P}_F^\rho(\mathcal{T})$ be the family retained after pruning. Its commitment regret under a signature objective C is

$$\text{Regret}_C(\mathcal{A}) = \inf_{P \in \mathcal{A}} C(\sigma_{\mathcal{T}}(P)) - \inf_{P \in \mathcal{P}_F^\rho(\mathcal{T})} C(\sigma_{\mathcal{T}}(P)). \quad (1.3)$$

The Safe-Commitment Law proves

$$\Gamma_{\mathcal{A}} = \Gamma_F^\rho(\mathcal{T}) \iff \text{every continuous monotone decision is preserved,} \quad (1.4)$$

$$\mathcal{K}_{\mathcal{A}} = \mathcal{K}_F^\rho(\mathcal{T}) \iff \text{every additive nonnegative-price decision is preserved.} \quad (1.5)$$

If the required equality fails, a decision in the corresponding class has strictly positive regret. The theorem therefore identifies the smallest observation object that remains invariant across every decision still open; all further reduction is a mathematical commitment to a smaller decision class.

Safe commitment is preservation of the geometry required by the future decision class.

Equivalently, discard a realization only when the future decisions in scope cannot detect that it is gone.

1.4 Theory architecture and principal results

The paper develops the theory in seven linked layers.

- C1. Semantic and resource contracts.** Terminalized computation, typed resource contracts, legal-realization semantics, and coherent hierarchy pushforward specify the domain before optimization begins.
- C2. Exact interface laws.** Contextual equivalence gives canonical componentwise-minimal directed states in a tree-summary model. Finite acyclic circuits and protocols translate under explicit accounting. Finite-field linear computations admit an exact interactive directional cut-rank orthant, even with nonlinear local encoders.
- C3. Conservation, carriers, and composition.** Deterministic carrier realizations coincide with zero-error network function codes. Temporal cut-state conservation and fractional packing yield bandwidth–time, area–time, and I/O projections under their declared service contracts. Monotonicity, local data processing, product additivity, and constructive synergy organize composition.
- C4. Decision-sufficient geometry.** Monotone observations reconstruct Γ ; additive prices reconstruct $\mathcal{K}$; the Safe-Commitment Law characterizes universal lossless retention; and explicit counterexamples show why marginal minima, one scalar, and universal convexification are insufficient.
- C5. Conditioned scaling.** A scale profile is derived from one legal hierarchical realization. Stable dilation limits, discrete hierarchy phase, positive-mode recursions, Perron–Floquet dynamics, arithmetic mode mixtures, interface reach, cap response, estimator bias, and finite-scale concentration give a unified calculus for Rent-like behavior.
- C6. Technology-conditioned design.** Technology enters as a capability, embedding, and cost system. Semialgebraic phase cells, robustness radii, certified inner/outer brackets, safe pruning, architecture–technology scale matching, link budgets, matrix multiplication, attention, state pooling, and layout-conversion laws turn CRG into a design calculus.
- C7. Integrated realization and validation.** A finite long-context Transformer registry exhibits technology-dependent class reversal. A working PyTorch FX representation constructs contracts, legal realization classes, typed obligations, scoped bounds, and decision certificates. A retention experiment tests the Safe-Commitment Law, and deterministic routed 2.5D/3D models carry one semantic ownership boundary into physical timing, shoreline, area, thermal, test, yield, and reliability coordinates.

1.5 Mathematical domain and extension space

The exact core is developed for finite deterministic computations because this class supports complete semantic quotients, exact zero-error converses, and fully auditable constructions. The CRG object itself is contract-parametric: randomized, approximate, stochastic, analog, reversible, noisy, and quantum realizations enter by enlarging the legal-realization semantics, signature coordinates, aggregation laws, and decision classes. Likewise, physical routing, timing, power delivery, thermal behavior, yield, reliability, and cost enter through technology contracts and embedding maps. The present results establish the common mathematical architecture on which those extensions compose.

2 Historical progression and relation to prior work

Computation-realization geometry lies at the intersection of theories that traditionally begin at different levels of abstraction. Computability fixes what can be calculated; information and communication theories characterize what can be represented or transmitted; automata and realization theories construct canonical internal state; VLSI and I/O theories establish physical or memory-service limits; contract and platform theories organize legal composition; compiler systems generate equivalent schedules and mappings; vector optimization and design-space exploration compare tradeoffs; and physical EDA embeds a selected logical object in a specific technology. CRG connects these layers through one computation-first object while preserving the different hypotheses and observables that make each predecessor powerful.

2.1 From computability to realization

Turing’s abstract machine separated effective computation from any particular hardware embodiment [20]. That abstraction made it possible to state machine-independent questions about computability and complexity. It also intentionally suppresses the physical organization that dominates modern systems: locality, communication paths, storage placement, hierarchy, energy, and technology capability. Gandy’s principles for mechanisms and Deutsch’s physical Church–Turing principle moved in the opposite direction: they asked which structural or physical assumptions make an abstract machine a model of realizable computation [65, 66]. CRG addresses a complementary question. Once a computation is declared, what are its legal physical realizations, what obligations do they jointly impose, and which representation of those alternatives remains sufficient for later decisions?

The word *realization* has a second established meaning in systems and control. Kalman’s realization theory reconstructs minimal state-space models from input–output behavior and characterizes the controllable and observable part of a linear dynamical system [67, 68]. CRG shares the concern with multiple internal embodiments of one external behavior, but its realization is broader: an algorithm, protocol, circuit, schedule, mapping, ownership assignment, hierarchy, resource allocation, and embedding that legally implements a declared computation. Minimal state dimension is one possible coordinate; the CRG object retains a family of jointly attainable resource and interface obligations.

Canonical-state ideas also arise in automata theory. Myhill–Nerode style quotients identify two histories when no future context can distinguish them [69]. The contextual quotients developed in Section 4 apply that principle to distributed terminal contexts: two assignments on one side of a hierarchy interface are equivalent exactly when every opposite-side context induces the same required outputs. The resulting state is a canonical semantic interface object rather than a machine-specific encoding.

Models of concurrent and streaming computation make communication part of the operational semantics. Kahn process networks, the Actor model, and synchronous dataflow represent computation through communicating processes, messages, and persistent local state [70, 71, 72]. They provide important realization classes inside CRG. CRG differs by ranging over multiple legal operational models and by placing their joint resource signatures inside a common decision-sufficiency calculus.

2.2 Information, communication, and decision sufficiency

Shannon’s theory separated source, channel, code, rate, and converse limit under a declared stochastic model [19]. It showed how an engineering field with many modulation and coding techniques could be organized around an implementation-independent object. CRG makes an analogous move for the passage from computation to physical realization: the theory first defines the legal realization family and only then asks which physical technology can serve and price its obligations. The difference is fundamental. Shannon could set message meaning aside for the channel-coding problem; CRG retains computational semantics because meaning determines which distinctions must cross which boundary.

Blackwell’s comparison of experiments provides a close decision-theoretic analogy [73]. One information structure is more informative than another when it is at least as valuable for every decision problem in a declared class. CRG similarly characterizes representations by the decisions they preserve. The upper obligation region Γ is sufficient and necessary for every continuous monotone resource decision, while the convex body $\mathcal{K}$ is sufficient and necessary for every additive nonnegative-price decision. The Safe-Commitment Law is therefore a design-sufficiency theorem: it identifies when a reduced realization family has exactly the same value for all decisions in the declared class.

Information complexity refines communication complexity by associating cost with information revealed during interaction and has yielded direct-sum and compression theorems [3, 4]. CRG can use such results as converses or coordinates inside a resource contract. Its geometric level is different: it records which algorithms, interactions,

storage policies, and architectures attain which joint obligations and which observation quotient is sufficient for the remaining physical decisions.

2.3 Communication complexity, distributed function computation, and network coding

Yao’s communication-complexity model asks how much parties with distributed inputs must exchange to compute a function [21]; subsequent work developed deterministic, randomized, interactive, multiparty, and information-complexity variants [22, 3]. These theories supply semantic lower bounds after the parties, input split, interaction rules, and output obligations have been declared. CRG registers those choices in ρ and places the converse beside constructive realizations and other resource coordinates.

Network information flow showed that coding at intermediate nodes can enlarge what a fixed carrier can achieve [23]. Network function computation extends that question from multicast identities to general functions and rate/load regions [24, 25]. This tradition contributes two central lessons to CRG: legal local transformations must be explicit, and the natural answer may be a region rather than a single capacity. CRG moves upstream of the fixed carrier by allowing ownership, hierarchy, architecture, and in some slices the carrier itself to vary across the legal realization family. Once a carrier is fixed, the carrier slice of CRG recovers deterministic zero-error network-function coding exactly.

2.4 VLSI area–time, I/O complexity, and parallel-machine models

Thompson’s area–time theory made geometric communication a source of circuit lower bounds in a two-dimensional VLSI model [26]. Hong and Kung’s red–blue pebble game similarly exposed unavoidable movement across a bounded fast-memory interface [5]. Communication lower bounds for numerical linear algebra and cache-oblivious constructions extended the program to multilevel and distributed memories [6, 8, 7]. These results are foundational converses: they establish what no legal realization inside a declared service model can evade.

BSP and LogP abstracted parallel machines through local work, communication, latency, and synchronization [27, 28]. Kung’s “communication geometry” classified the intermodule network structure of a selected parallel algorithm [1]. CRG preserves these service and machine models as named projections. The additional step is to place them inside a larger family in which algorithm, schedule, storage, replication, interaction, and hierarchy may change before the machine or technology is fixed.

2.5 Rent’s rule, graph structure, and multiscale measurement

Landman and Russo’s pin-versus-block relation made multiscale connectivity a practical circuit descriptor [29]. Donath linked connectivity statistics to placement and average interconnection length; subsequent work developed intrinsic and heterogeneous Rent parameters, local variation, range structure, estimator conventions, and systematic wire-length prediction [82, 83, 84, 85, 30, 86]. Hypergraph partitioning, cut metrics, tree widths, branch widths, and separator methods provide additional structural projections of a realized graph [52, 53, 81].

A Rent profile is conditioned on a realization, hierarchy, terminal convention, accounting rule, estimator, and scale window. CRG therefore places it after the realization map $\sigma\tau$, not before the legal family is defined. The scaling theory in this paper unifies several mechanisms that can produce Rent-like observations: exact scale covariance, regular variation, discrete hierarchy phase, positive-mode recursions, heterogeneous mode curvature, interface reach, terminal saturation, and estimator bias. The result is a generative theory of conditioned scale profiles rather than one portable exponent detached from its measurement contract.

2.6 Platforms, interfaces, contracts, and design rules

Platform-based design and orthogonalization of concerns separate functional specification, architecture, and implementation across reusable layers [31, 32]. Interface automata and contract-based design formalize assumptions, guarantees, compatibility, refinement, and composition [33, 34]. These traditions supply the legal and compositional discipline used by CRG: every theorem is indexed by a computation, resource contract, hierarchy, and accounting rule, and every structure map must preserve that contract.

Semiconductor design rules and process design kits provide the industrial analogue. They condense coupled optical, chemical, process, integration, variability, and reliability knowledge into executable constraints and models, allowing design to proceed at a higher level without discarding the physical process window. Mead–Conway design

methodology made this abstraction a central mechanism for managing VLSI complexity [87, 88]. CRG extends the same architectural idea to computation-to-technology design: the contract defines legal transformations, the geometry records their joint obligations, and the Safe-Commitment Law determines which alternatives can be removed without changing the remaining decisions.

2.7 Set-based design, tradespaces, and systems architecture

A parallel design tradition approached premature commitment from the engineering-process side. Set-based concurrent engineering retains families of feasible alternatives, communicates sets across disciplines, and narrows those sets as knowledge accumulates rather than iterating one point solution through serial rework [96, 97]. Multi-attribute tradespace exploration maps broad architecture families against stakeholder value and cost and studies how preferences move as context changes [98]. Model-based systems engineering provides explicit requirements, interface, behavior, verification, and lifecycle relations across system levels [99].

CRG gives these set-retention practices a computation-theoretic object and an exact narrowing criterion. The initial set is not an untyped collection of concepts; it is the legal family generated by a terminalized computation and a resource contract. Each member carries a typed joint signature and, where available, a constructive witness or an outer obligation. The Safe-Commitment Law then characterizes the precise equality that permits the family to be narrowed while preserving every future decision in a declared class. Set-based practice supplies an organizational precedent for delayed commitment; CRG supplies the realization semantics, decision-sufficient geometry, and proof of when convergence is lossless.

2.8 Program transformations, intermediate representations, and equality saturation

Parnas’s program-family formulation made commonality and variation across related programs a first-class design object rather than an after-the-fact catalogue of implementations [100]. Compiler research has long separated semantic computation from legal transformations and schedules. The polyhedral model represents affine loop nests and enables legality-preserving transformations for locality and parallelism [75]. Halide makes the algorithm–schedule split explicit [76]; TVM combines graph-level and operator-level optimization with autotuning [77]; and MLIR provides extensible, multilevel intermediate representations that preserve structure across lowering stages [78].

Equality saturation addresses the phase-ordering problem by retaining many equivalent programs in an e-graph before extracting one according to a cost model [79, 80]. Semantic e-graphs now carry this non-destructive representation into EDA by joining logic optimization and technology mapping across multiple semantic levels [101]. This is a close constructive relative of CRG: both delay premature selection and preserve alternatives through a shared representation. Equality saturation is centered on equationally equivalent program expressions and extraction from a rewrite closure. CRG additionally requires terminal semantics, resource-contract legality, hierarchy-indexed typed obligations, problem-specific converses, observation-sufficient quotients, technology capabilities, and proof-governed commitment.

Abstract interpretation provides a general theory of sound semantic approximation through ordered domains and fixpoint abstraction [74]. CRG shares the concern with what an abstraction preserves, but its central theorem is decision-oriented rather than solely safety-oriented: Γ and $\mathcal{K}$ are characterized by the complete classes of optimization queries they answer exactly. This yields an explicit boundary between reversible abstraction and commitment regret.

2.9 Vector optimization, Pareto sets, and robust decisions

Multiobjective and vector optimization study feasible images, ordering cones, upper images, scalarizations, support functions, and inner/outer approximations [2, 89, 35, 36]. Approximate Pareto sets address the computational size of tradeoff frontiers [90]. Robust optimization studies decisions that remain valid over parameter uncertainty sets [91]. CRG uses these mathematical tools but ties each image to a semantic realization fiber and a declared observation class. The exact region $\mathcal{R}$, upper region Γ , and convex body $\mathcal{K}$ are not interchangeable summaries: each is proved sufficient for a different family of future decisions.

Robust and parametric optimization ask how decisions change under uncertainty or parameter variation. CRG’s semialgebraic phase cells, full-label robustness radii, and argmin-hitting retention provide a realization-aware parametric calculus. The decisive distinction is between uncertainty that reprices an unchanged legal family and a capability change that enlarges or alters that family.

2.10 Architecture exploration, compiler mapping, and physical EDA

Modern design-space exploration systems search large spaces of dataflows, loop schedules, mappings, memories, accelerator structures, distributed parallel plans, and chiplet organizations [37, 38, 39, 40, 17, 18, 41]. Recent multi-level and chiplet work widens the representation itself: MLDSE introduces recursively composable hardware and mapping intermediate representations, while Mozart co-designs reusable chiplet ecosystems and accelerators with physical validation [92, 93]. Architecture-level estimators and physical flows supply energy, area, timing, and placement information [43, 42]. Open3DBench extends reproducible physical evaluation to partitioning, placement, routing, RC extraction, and thermal analysis for 3D-ICs [94]. These systems are constructive engines for the left and right sides of CRG: compilers and DSE generate legal witnesses, while physical tools evaluate embeddings and return improved witnesses.

The CRG contribution is the intervening scientific contract. It specifies how the candidate space is constituted, retains realization identity where a signature is insufficient, attaches typed obligations and outer bounds, identifies which reduced representation preserves which decision class, and records when a technology change is a repricing event or a change of legal space. In this sense, design-space exploration searches a map; CRG defines the semantic rules for drawing, reducing, and reopening that map.

2.11 Physical realization, reversibility, and thermodynamic cost

Landauer related logical irreversibility to physical dissipation [45], and Bennett developed reversible computation and the thermodynamic consequences of retaining or erasing computational history [95]. Physical-computation formalisms study abstraction, composition, and multiple realizability across substrates [44]. CRG provides a compatible realization layer. Logical reversibility, heat, noise, error correction, and stochastic work can be introduced as legal primitives, state coordinates, and technology observations. The finite deterministic theory developed here supplies the semantic and geometric base on which those physical extensions are defined.

2.12 Comparative synthesis

Table 2 summarizes where the principal traditions begin and what they return. The boundaries are methodological, not hierarchical: each theory is exact for the question it asks. CRG supplies the common pre-commitment object that allows their outputs to be composed without silently changing quantifiers or mixing incompatible optima.

Table 2: Relation of CRG to neighboring theories and design methods.

Tradition	Primary object	Usually fixed before analysis	Role inside CRG
Computability	Abstract machine and computable function	Machine model up to simulation equivalence	Defines the semantic task independent of one physical embodiment
Information theory	Source, channel, code, rate/capacity	Source/channel law and fidelity criterion	Supplies coding concepts, converses, and the precedent for implementation-independent engineering theory
Communication complexity	Function, parties, cut, protocol class	Input ownership and interaction model	Supplies semantic cut obligations and interactive converses
Network coding	Carrier graph and code region	Network, sources, demands, alphabet model	Supplies attainable carrier-load regions and coding transformations
VLSI and I/O theory	Layout or memory-service model	Physical dimension, DAG, memory hierarchy, charging rule	Supplies scoped area-time, bandwidth-time, and movement lower bounds
Rent and graph structure	Realized graph under a hierarchy	Netlist, partition, estimator, terminal convention	Supplies conditioned multiscale diagnostics of one realization
Contracts and platforms	Assumption/guarantee interface and platform family	Component boundaries and composition semantics	Supplies legality, refinement, and reusable interface discipline

Tradition	Primary object	Usually fixed before analysis	Role inside CRG
Set-based design and tradespaces	Feasible concept sets and value/cost spaces	Candidate-generation rules, attributes, and stakeholder model	Supplies delayed convergence and lifecycle-change reasoning; CRG supplies semantic legality and exact retention conditions
Compiler IR and equality saturation	Equivalent programs, schedules, rewrites	Rewrite system, source semantics, extraction cost	Supplies constructive realization generation and delayed extraction
Vector optimization	Feasible image and ordering cone	Candidate image and objective class	Supplies upper images, support duality, and approximation theory
DSE and EDA	Registered candidates and physical embeddings	Representation, candidate generator, technology model, objective	Supplies witnesses, search, and physical evaluation
CRG	Legal realization bundle and decision-sufficient geometry	Computation and resource contract only	Constitutes the pre-technology design family, composes converses and witnesses, and proves when reduction is lossless

2.13 The gap closed by computation-realization geometry

The preceding literatures collectively contain nearly every ingredient needed for a theory of computation realization, but they distribute those ingredients across different starting points and quantifier orders. CRG closes the gap by joining six properties in one formal object:

- (i) the computation and its terminal semantics precede every physical projection;
- (ii) a resource contract defines the legal realization family before a candidate registry is searched;
- (iii) each realization carries a typed joint signature rather than a single communication scalar;
- (iv) hierarchy changes act through coherent structure maps;
- (v) observation classes determine exact decision-sufficient quotients and therefore exact safe-commitment conditions; and
- (vi) technology enters after the legal geometry as capability, embedding, and cost, with explicit repricing and reopen semantics.

This synthesis is the theoretical advance: a communication theory whose object is the realization of computation and whose central law governs what may be forgotten before physical commitment.

Sequence and nonhomogeneous-product lineage. The sequence-level regular-variation results sit in the classical theory of regularly varying sequences [102, 103]. The positive-product mechanism is related to weak ergodicity and nonhomogeneous products of nonnegative matrices [104, 105, 106, 107].

Performance and accelerator co-design lineage. Roofline provides the canonical fixed-kernel compute-bandwidth balance [108]; Stream and Simba provide neighboring layer-fused and multi-chip-module accelerator systems [109, 110]. CRG retains these as conditioned projections or realizations inside a prior legal-realization object.

Part I

Semantic contracts and exact resource geometry

3 Terminalized computation and resource contracts

The mathematical object to be realized is separated from the resources by which it is realized. This separation fixes the quantifiers used throughout: first a semantic task, contract, and hierarchy are fixed; only then are legal realizations and their interface signatures compared.

Definition 3.1 (Terminalized deterministic computation). *A terminalized deterministic computation is a map*

$$F : \mathcal{X} = \prod_{i \in I} X_i \longrightarrow \mathcal{Y} = \prod_{j \in O} Y_j,$$

where every alphabet is finite. A hierarchy consists of a finite tree $\mathcal{T} = (V, E)$ and placement maps $\lambda_I : I \rightarrow V$ and $\lambda_O : O \rightarrow V$. Vertex v owns the local input alphabet

$$X_v = \prod_{i: \lambda_I(i)=v} X_i$$

and must produce the local output alphabet

$$Y_v = \prod_{j: \lambda_O(j)=v} Y_j.$$

Definition 3.2 (Core resource contract). *A core deterministic resource contract is a tuple*

$$\rho = (\mathcal{B}, S, D, M, R, \varepsilon, \Lambda, \chi),$$

where $\mathcal{B}$ is the allowed local primitive basis, S is a local-work or circuit-size budget, D a causal-depth budget, M a storage budget, R a replication or recomputation budget, ε an allowed error (zero in the exact results below), Λ the admissible hierarchy and terminal-placement rules, and χ the communication-accounting type. A ρ -legal realization P is a finite hierarchical protocol or circuit satisfying these typed bounds. Contract relaxation is semantic inclusion: $\rho \preceq \rho'$ exactly when every ρ -legal realization is ρ' -legal.

To make hierarchy contraction formal, each scalar bound is represented by a finitely supported charge on hierarchy sites or interfaces together with a declared associative aggregation law. Local work and storage use addition; retained interface traffic and replication charges use addition over surviving interfaces; causal depth and protocol rounds use max-plus path composition; terminal placement uses direct image; exact error remains zero, while an approximate extension must declare an associative error-composition operation $\oplus_\varepsilon$; and χ is preserved. A connected-fiber contraction $q : \mathcal{T} \rightarrow \mathcal{T}'$ pushes a charge forward by aggregating all charges in each fiber and on each induced surviving interface.

A resource budget is a legality constraint on the complete realization; a resource charge is the additive or max-plus object pushed through a hierarchy map. A budget need not equal the sum of local charges unless the contract declares that aggregation rule. Thus, for example, a package-wide peak or shared memory pool is not counted once per stage merely because stage-local timing expressions are written separately. Extensions may add stochastic, analog, reversible, or technology-specific coordinates, but they must specify legal-realization semantics, an associative aggregation law, and pushforward behavior before the coherence results apply.

3.1 How a resource contract changes the realization family

The contract is part of the mathematical problem, not metadata attached after optimization. Three simple changes illustrate its effect.

Time sharing.

Two stage-local state requirements m_1, m_2 aggregate as $m_1 + m_2$ when dedicated engines coexist and as $\max\{m_1, m_2\}$ when one state pool is reused across disjoint time intervals. The latter realization also carries a schedule and utilization coordinate. Replacing one aggregator by the other changes legality.

Replication and recomputation.

A replicated table or recomputed intermediate can reduce communication while increasing work, state, or replication charges. Those are different legal points in one typed geometry, not different estimates of one fixed graph.

Approximation.

Permitting error $\varepsilon > 0$ enlarges the legal family only after the correctness relation and the composition law $\oplus_\varepsilon$ have been declared. Approximate and exact geometries are different slices of the theory.

The hierarchy rule Λ is equally active. A partition can be a nested accounting cut, a physical carrier, a clock or ownership boundary, or a measurement hierarchy. The contract specifies which interpretation is in force and which resource charges survive contraction. This is the mechanism that prevents a static cut count from being silently read as bandwidth, latency, routed congestion, or energy.

Proposition 3.3 (Functoriality of contract pushforward). *For every core contract with the aggregation laws of Theorem 3.2, connected-fiber contractions satisfy*

$$\text{id}_* \rho = \rho, \quad (r \circ q)_* \rho = r_*(q_* \rho).$$

Proof. For an additive local charge a , the charge assigned to a coarse site w by the composite contraction is

$$((r \circ q)_* a)(w) = \sum_{v: (r \circ q)(v)=w} a(v) = \sum_{u: r(u)=w} \sum_{v: q(v)=u} a(v) = (r_*(q_* a))(w).$$

The same regrouping proves the claim for additive interface and replication charges. Max-plus path aggregation is associative because path concatenation and max are associative; direct image of terminal placement is functorial; exact error is fixed at zero; and every admitted approximate coordinate is functorial by the required associativity of $\oplus_\varepsilon$. The primitive basis and accounting type are unchanged. Identity maps leave every support and charge unchanged, so componentwise equality gives both identities. $\square$

Definition 3.4 (Typed interface signature). *For an oriented hierarchy interface e , a typed signature is*

$$\mu_e = (s_e, \nu_e, t_e, d_e, a_e, \pi_e),$$

where s_e is bits per state or message, ν_e is message events per operation, $t_e = s_e \nu_e$ is bits per operation, d_e is a causal-round or depth contribution, a_e is an activity factor, and π_e is an optional traffic or burst distribution. An analysis may project to any declared subset. Static cut accounting uses s_e or a fixed-bit width; throughput and link calculations use t_e ; latency uses d_e ; and dynamic energy requires t_e together with a_e and a technology model. These coordinates are not interchangeable.

Definition 3.5 (Interface-accounting convention). *For a finite message alphabet Σ , define its semantic log-cardinality and fixed-length binary width by*

$$\ell_{\text{sem}}(\Sigma) = \log_2 |\Sigma|, \quad \ell_{\text{bit}}(\Sigma) = \lceil \log_2 |\Sigma| \rceil.$$

For a sequence of message events, capacities add event by event. The contract records $\chi \in \{\text{sem}, \text{bit}\}$, or an explicitly specified richer coding rule. Semantic capacity is exact for state-space and product identities; fixed-bit traffic is the physical convention for Boolean circuit nets. The two conventions coincide at one bit for a binary alphabet.

Unless stated otherwise, the event schedule is fixed independently of the input, conditional absence is charged as an alphabet symbol, and opposite directions are charged separately. The event sum is provisioned alphabet capacity, not Shannon entropy of a reachable-transcript distribution. Joint coding across events, expected variable length, compression, retransmission, broadcast cost, and link sharing require a richer χ stating decoder, timing, and charging rules. Bits per operation alone do not specify instantaneous bandwidth or latency; event-rate, burst, and service coordinates are also required.

4 Canonical distributed interface states

Removing an undirected edge $\{u, v\}$ creates two components. Write $V_{u|v}$ for the component containing u , $X_{u|v}$ for the product of inputs placed there, and $Y_{v|u}$ for the outputs placed on the opposite side.

Definition 4.1 (Separable directed-summary realization). *For every oriented edge $u \rightarrow v$, choose a finite message alphabet $\Sigma_{u \rightarrow v}$. A separable directed-summary realization consists of local maps*

$$\delta_{u \rightarrow v} : X_u \times \prod_{w \in N(u) \setminus \{v\}} \Sigma_{w \rightarrow u} \longrightarrow \Sigma_{u \rightarrow v}$$

and output maps

$$\eta_u : X_u \times \prod_{w \in N(u)} \Sigma_{w \rightarrow u} \longrightarrow Y_u.$$

Because the graph is a tree, the message on $u \rightarrow v$ is recursively a function only of $X_{u|v}$. Oppositely directed messages summarize their own sides and do not depend on one another. This is a noninteractive two-pass summary model, not the general interactive model introduced in Section 5.

Definition 4.2 (Directed contextual equivalence). *For $a, a' \in X_{u|v}$, define*

$$a \equiv_{u \rightarrow v} a' \iff F_{Y_{v|u}}(a, z) = F_{Y_{v|u}}(a', z) \text{ for every } z \in X_{v|u}.$$

Let

$$Q_{u \rightarrow v} = X_{u|v} / \equiv_{u \rightarrow v}, \quad \kappa_{u \rightarrow v} = \log_2 |Q_{u \rightarrow v}|.$$

The real logarithm is the semantic capacity; a fixed-length binary encoding uses $\lceil \kappa_{u \rightarrow v} \rceil$ bits.

Theorem 4.3 (Canonical directed-summary theorem). *For every finite terminalized computation F and every tree hierarchy $\mathcal{T}$:*

- (a) *the directed contextual classes are locally composable;*
- (b) *the class messages $Q_{u \rightarrow v}$ yield a separable directed-summary realization of all outputs;*
- (c) *every separable directed-summary realization satisfies $|\Sigma_{u \rightarrow v}| \geq |Q_{u \rightarrow v}|$ on every oriented edge; and*
- (d) *if equality holds on every oriented edge, the realization is the canonical quotient realization on every reachable local input–message tuple, up to independent renaming of message states; behavior on unreachable tuples is unconstrained.*

Thus

$$\kappa_{\mathcal{T}}^{\leftrightarrow}(F) = (\kappa_{u \rightarrow v})_{u \rightarrow v}$$

is an implementation-independent, encoding-optimal, jointly realizable directed-capacity vector for the hierarchy.

Proof. Fix $u \rightarrow v$. The component $V_{u|v}$ is the disjoint union of u and the components $V_{w|u}$ for $w \in N(u) \setminus \{v\}$. Suppose $a_w, a'_w \in X_{w|u}$ satisfy $a_w \equiv_{w \rightarrow u} a'_w$ for every such neighbor, and fix the same local assignment at u . For any assignment z on $V_{v|u}$, replace the a_w by a'_w one component at a time. By definition, each replacement preserves every output outside $V_{w|u}$ for every assignment to the remaining variables, hence in particular every output in $V_{v|u}$. The combined assignments are therefore equivalent under $\equiv_{u \rightarrow v}$. The outgoing class is consequently a well-defined function of the local input and incoming classes.

For an output at vertex u , choose arbitrary representatives of the incoming classes. Replacing any representative by another in the same class preserves every output outside that neighbor component, including the outputs at u . The local output is therefore well defined and equals F .

For minimality, consider any separable realization. Its message $m_{u \rightarrow v}$ is a deterministic function of the assignment on $V_{u|v}$. If inequivalent assignments a, a' produced the same message, some opposite-side assignment z would distinguish an output in $V_{v|u}$. Every opposite-side state would be identical in the two runs, and the only dependence on the u -side would be through the identical message. Determinism would force identical outputs, a contradiction. Thus different contextual classes require different messages.

If every alphabet has minimum cardinality, its message map can neither merge two classes nor use spare states to split one class. It induces a bijection $Q_{u \rightarrow v} \rightarrow \Sigma_{u \rightarrow v}$, and the local maps are conjugate to the quotient maps under these independent bijections on every reachable local input–message tuple. Their values on unreachable tuples may be assigned arbitrarily without changing correctness, minimum message cardinality, or canonical equivalence on the reachable domain. $\square$

Interpretation. The theorem is simultaneously constructive, minimal, and joint. It does not obtain one lower bound per edge and then splice incompatible edgewise optima. The same quotient realization attains every directed minimum at once. The message state on $u \rightarrow v$ is the complete set of distinctions on the u -side that can still matter to an output on the v -side under some legal opposite-side context. Any smaller alphabet merges two distinguishable contexts; any larger alphabet contains representation redundancy.

The result also explains why terminal placement is load bearing. Moving an output to another vertex changes which contexts can distinguish two assignments and therefore changes the quotient itself. A “communication requirement of the function” is consequently incomplete until input and output ownership have been declared.

Example 4.4 (Three-node worked quotient). *Let a path $u - v - w$ carry input bits x at u and y at w , with output $x \oplus y$ required at v . Across $u \rightarrow v$, the two values of x are contextually inequivalent because fixing $y = 0$ distinguishes the required output; hence $Q_{u \rightarrow v} \cong \{0, 1\}$. Symmetrically, $Q_{w \rightarrow v} \cong \{0, 1\}$. In the reverse directions no output lies beyond the edge, so the quotient has one state. The canonical realization sends one bit inward from each leaf and computes parity locally at v . The directed capacity vector follows from semantics alone and is attained without choosing a binary encoding convention in advance.*

Corollary 4.5 (Fixed-bit canonical widths). *Under fixed-length binary accounting,*

$$\hat{\kappa}_{u \rightarrow v} = \lceil \kappa_{u \rightarrow v} \rceil$$

is jointly realizable and componentwise minimal among all fixed-length encodings in the separable directed-summary model. Each ceiling adds less than one bit beyond semantic capacity.

Corollary 4.6 (Single-root quotient). *If all outputs are placed at a root r , every edge directed toward r has the contextual-equivalence state space of its subtree, while every reverse edge has one state. The canonical realization is bottom-up and is analogous to a Myhill–Nerode quotient for the selected variable tree.*

Corollary 4.7 (Independent-bit routing). *Let source bit x_i be placed at one vertex and copied to an arbitrary set of output terminals. For oriented edge $u \rightarrow v$, let $k_{u \rightarrow v}$ be the number of source bits located in $V_{u|v}$ with at least one requested copy in $V_{v|u}$. Then*

$$|Q_{u \rightarrow v}| = 2^{k_{u \rightarrow v}}, \quad \kappa_{u \rightarrow v} = k_{u \rightarrow v}.$$

Consequently, total directed semantic capacity across an undirected edge equals the number of independent routed nets separated by that cut.

Proof. The opposite-side outputs depend on exactly those $k_{u \rightarrow v}$ source bits. Two assignments on the u -side are contextually equivalent precisely when they agree on all of them, giving $2^{k_{u \rightarrow v}}$ classes. $\square$

Example 4.8 (Broadcast and all-reduce). *A single source bit broadcast to many consumers has one bit of semantic capacity on every directed edge separating the source from at least one consumer and zero in the reverse direction. One bit of information can thus accumulate arbitrarily large multiscale incidence. By contrast, if every node must output the parity of all source bits, each side of every nontrivial edge has two summary states (its local parity), so one bit is required in each direction. Semantic information and physical distribution incidence are therefore distinct quantities.*

5 General causal realizations and exact certificates

The quotient theorem is exact within the separable two-pass model. A general circuit or protocol may cross the same interface in several causal stages, so the realization class must be enlarged before making interactive claims. We use four explicit conventions: circuits are finite, combinational, and acyclic; each Boolean signal is one driver-to-sinks hyperedge net; gates and terminals are placed at hierarchy vertices; and each hierarchy edge is either a nested cut-accounting boundary or a physical tree hop, as declared by ρ . Net incidence counts one bit per crossed tree edge, not one bit per sink pin. Sequential reuse, registers, variable schedules, routed congestion, and nonfree branching require additional time, storage, or carrier coordinates.

The section makes three successive enlargements of the realization model. First, a circuit is represented as a causal protocol so that static net incidence and message traffic can be compared under one accounting contract. Second, interaction is admitted, showing precisely how nonlinear computations can alter the lower bounds derived from one-way contextual states. Third, a temporal service window is introduced, allowing retained state and communication to enter one conservation law. The three steps separate representation, interaction, and time instead of folding them into one overloaded “communication cost.”

For each enlargement the proof pattern is the same:

$$\text{semantic range or quotient} \implies \text{outer obligation} \quad \text{and} \quad \text{explicit protocol or circuit} \implies \text{inner witness}.$$

When the two meet, the corresponding CRG slice is exact.

5.1 Finite causal protocols and circuit translations

Definition 5.1 (Finite acyclic tree protocol). *A finite acyclic protocol P on a tree hierarchy consists of a finite event DAG. Each message event a is labeled by an oriented tree edge, outputs a symbol in a finite alphabet Σ_a , and is a deterministic function of the local input at its source vertex and earlier messages received there. A conditional or absent message is an ordinary alphabet symbol, so the event schedule is static. Each output event is similarly a deterministic function of local input and earlier received messages. The protocol computes F when all terminal outputs equal $F(x)$ for every x .*

For tree edge e , define semantic alphabet traffic and fixed-length binary traffic by

$$a_e(P) = \sum_{a: a \text{ labeled by } e} \log_2 |\Sigma_a|, \quad c_e(P) = \sum_{a: a \text{ labeled by } e} \lceil \log_2 |\Sigma_a| \rceil.$$

Write $t_e^x(P) = a_e(P)$ under semantic accounting and $t_e^x(P) = c_e(P)$ under fixed-bit accounting. A round-resolved signature may retain every message event separately; $t^x(P)$ is its declared static projection.

Definition 5.2 (Circuit cut vector). Let C be a finite Boolean circuit with one hyperedge net per signal, and let $\mathcal{T}$ place every gate and terminal at a tree vertex. For tree edge e , let M_e denote either side of its cut and define

$$b_e(C, \mathcal{T}) = |\{n : \text{net } n \text{ has pins on both sides of } e\}|.$$

The vector $b(C, \mathcal{T})$ is the multiscale cut-incidence vector of the placed implementation.

The translation treats a gate or terminal at v as a local deterministic event, a one-bit net as one-bit messages on its minimal connecting subtree, and fanout as replication at a tree branch. Replication is free only in the cut-only contract; otherwise it contributes local work, storage, and possibly depth. Topological gate order becomes event-DAG order. A k -symbol message uses a chosen injective $\lceil \log_2 k \rceil$ -bit code whose encoder and decoder logic is included in the local event circuits. Reuse through time is either unrolled into distinct events or represented by stored sequential state and therefore cannot be inferred from a static cut vector.

Theorem 5.3 (Circuit-to-protocol factorization). For every finite acyclic Boolean circuit C and tree placement $\mathcal{T}$ of its gates and terminals, there is a finite acyclic tree protocol computing the same terminalized function with

$$c_e(P) = b_e(C, \mathcal{T})$$

for every tree edge e , under a communication-only contract in which forwarding, replication, and multicast at tree branch vertices are free. The equality is an undirected cut-total identity; the net driver determines the direction on every crossed edge. Moreover,

$$\sum_{e \in E(\mathcal{T})} b_e(C, \mathcal{T}) = \sum_{n \in E(C)} |E(\text{Steiner}_{\mathcal{T}}(\text{pins}(n)))|,$$

where the right side counts edges in the minimal subtree connecting the pins of each circuit net.

Proof. Process circuit nets in topological order of their drivers. Once the driver of net n is known, route its one-bit value along the minimal subtree of $\mathcal{T}$ connecting the driver and all sinks, replicating at branch vertices. A tree edge is used by n exactly when deleting that edge separates at least two pins of n , equivalently when n crosses the associated circuit cut. Exactly one bit for n traverses each such edge, proving the pointwise and summed identities. A net is routed only after its driver is computed, so the event schedule is acyclic. $\square$

Remark 5.4 (Resource cost of materialized forwarding). If forwarding is materialized as logic and every physical tree hop consumes a causal stage, the construction adds $O(\sum_e b_e)$ forwarding events and at most a factor $\text{diam}(\mathcal{T})$ to path depth. If hierarchy edges are nested accounting boundaries, the theorem is a cut representation and adds no logical depth. The contract must choose the interpretation.

Theorem 5.5 (Protocol-to-circuit compilation). Let P be a finite acyclic tree protocol. Choose an injective fixed-length binary code for every message alphabet, and suppose each encoded message or output event, including required encoder and decoder logic, is supplied by a Boolean circuit over $\mathcal{B}$ with size s_a and depth d_a as a function of its locally available bits. Then P compiles into a Boolean circuit computing the same terminalized function with

$$S(C) \leq \sum_a s_a,$$

and depth at most the maximum weighted length of a path in the event DAG, where event a has weight d_a . Every message bit becomes a circuit net crossing the corresponding tree interface, hence

$$b_e(C, \mathcal{T}) \leq c_e(P).$$

Combinational unrolling represents a message reused in several causal rounds by separate nets; a sequential realization may instead trade storage and time for wires under an enriched contract.

Proof. Topologically order the event DAG. Instantiate the stated circuit for each event and connect its input pins to local input terminals and the earlier message outputs on which it depends. Acyclicity gives a legal circuit. Summing local event sizes proves the size bound, and the longest weighted event path proves the depth bound. Each encoded message bit is one net crossing its labeled interface, which proves the cut bound. $\square$

What the two translations establish. The circuit-to-protocol direction preserves the exact multiscale incidence of one placed combinational realization under the cut-only contract. The protocol-to-circuit direction records the local logic and causal depth needed to implement arbitrary finite message alphabets. Together they show that “circuit” and “protocol” are two representations of realizations inside CRG, not competing definitions of the design space. Their signatures agree only after the contract specifies whether forwarding, fanout, storage, and time are free or charged.

As a simple example, a one-bit signal driven at one leaf and consumed at two other leaves uses each edge of the minimal connecting subtree once. Its summed circuit-cut incidence equals the number of edges in that subtree; the protocol witness routes the bit along exactly those edges and replicates only at their branch point. If branch replication is charged, the same semantic realization acquires an additional local-work or activity coordinate without changing the cut identity.

Corollary 5.6 (Representation equivalence with explicit overhead). *Circuits and finite acyclic hierarchical protocols are mutually translatable. The forward translation exposes multiscale cut incidence exactly under the communication-only contract; the reverse compilation preserves semantics with explicit local-computation and causal-depth overhead. Arbitrary interface encoding and interaction therefore belong inside the feasible realization family rather than being excluded by definition.*

5.2 Where interaction changes the signature

Example 5.7 (INDEX interaction boundary). *Alice holds $x \in \{0, 1\}^n$, Bob holds $i \in [n]$, and Bob must output x_i . A one-way Alice-to-Bob summary must distinguish all 2^n strings, so its semantic capacity is n bits. A two-round protocol lets Bob send i using $\lceil \log_2 n \rceil$ bits and Alice return x_i using one bit. Thus*

$$D_{\text{one-way}}^{A \rightarrow B}(\text{INDEX}_n) = n, \quad D_{\text{interactive}}(\text{INDEX}_n) \leq \lceil \log_2 n \rceil + 1.$$

The legal directional signatures $(n, 0)$ and $(1, \lceil \log_2 n \rceil)$ can be selected by different price vectors. With uniform bit prices and $n \geq 4$, the interactive realization wins; if Bob-to-Alice traffic is sufficiently expensive, the one-way realization wins. A one-way contextual summary, total communication, and a technology-weighted interface objective are different observations.

In the noninteractive model, contextual quotients are both constructive points and matching lower bounds. In a general model, a semantic outer bound requires a problem-specific converse from communication complexity, information complexity, algebra, or another appropriate method. The operational discipline is therefore to retain certified brackets

$$\mathcal{R}_{F,\text{in}}^\rho(\mathcal{T}) \subseteq \mathcal{R}_F^\rho(\mathcal{T}) \subseteq \mathcal{R}_{F,\text{out}}^\rho(\mathcal{T})$$

and declare every gap.

5.3 Complete interactive finite-field linear orthant

Theorem 5.8 (Linear contextual states are cut ranks). *Let $F(x) = Mx$ over $\text{GF}(2)$, split inputs as $x = (x_A, x_B)$, and place a set of outputs Y_B on side B . If $M_{B,A}$ is the submatrix mapping x_A to outputs required on B , then $a, a' \in \text{GF}(2)^A$ are contextually equivalent for $A \rightarrow B$ exactly when*

$$M_{B,A}(a - a') = 0.$$

The quotient has $2^{\text{rank } M_{B,A}}$ classes, and

$$\boxed{\kappa_{A \rightarrow B} = \text{rank}_{\text{GF}(2)} M_{B,A}.$$

Proof. For any outside assignment b , the outputs on B differ by $M_{B,A}(a - a')$. They agree for every b exactly when this vector is zero. The number of cosets of $\ker M_{B,A}$ is $2^{\text{rank } M_{B,A}}$. $\square$

Lemma 5.9 (Interactive directional range bound). *Delete one tree edge and call the sides A and B . In any exact deterministic finite causal protocol with a static charged event schedule,*

$$t_{A \rightarrow B} \geq \max_b \log_2 |\{F_B(a, b) : a \in X_A\}|$$

under semantic alphabet accounting; fixed binary width rounds the right side up.

Proof. Fix b and retain the ordered tuple of every A -to- B symbol. Given that tuple and the receiver input, every receiver-local event and reverse message is determined recursively by induction over the event DAG. Equal forward tuples must therefore give equal required receiver outputs. The tuple has at least the displayed number of values and at most the product of its event alphabet cardinalities. $\square$

Lemma 5.10 (Local composition of linear quotient states). *For a component C exposed by deleting a tree edge, let*

$$L_C = M_{O_{\bar{C}}, I_C}, \quad Q_C = \mathbb{F}_{\mathfrak{q}}^{I_C} / \ker L_C,$$

where $\mathfrak{q}$ is the field cardinality. If $O_B \subseteq O_{\bar{C}}$, then

$$\ker M_{O_{\bar{C}}, I_C} \subseteq \ker M_{O_B, I_C}.$$

Consequently $M_{O_B, I_C} x_C$ is a well-defined linear function of Q_C . If disjoint incident components $C_1, \dots, C_s$ meet at v and O_B lies outside every C_i , their aggregate contribution to O_B is a linear function of x_{I_v} and the states Q_{C_i} .

Proof. Deleting output rows from $M_{O_{\bar{C}}, I_C}$ can only enlarge its kernel. Thus

$$L_C x_C \mapsto M_{O_B, I_C} x_C$$

is well defined on $\text{im } L_C \cong Q_C$. Apply this factor map to each incident component and add the resulting vectors together with the local term $M_{O_B, I_v} x_{I_v}$. $\square$

Definition 5.11 (Interactive linear traffic contract). *Let $F(x) = Mx$ over a fixed finite field $\mathbb{F}_{\mathfrak{q}}$, with input coordinates and required output rows placed on $\mathcal{T}$. For an oriented edge $e = A \rightarrow B$, put*

$$M_e = M_{O_B, I_A}, \quad r_e = \text{rank}_{\mathbb{F}_{\mathfrak{q}}} M_e.$$

The contract admits exact deterministic finite causal tree protocols, arbitrary deterministic local maps (not necessarily linear), and a static charged event schedule. Each vertex begins with only its placed input and fixed input-independent advice; no input-dependent state is prepositioned across a cut. Traffic is charged by [Theorem 3.5](#); local work, storage, rounds, and depth are unrestricted in this slice. Conditional absence remains an ordinary charged symbol.

Theorem 5.12 (Exact interactive linear lower corner and orthant). *Under [Theorem 5.11](#), arbitrary interaction and nonlinear local encoding cannot reduce the directional load below $r_e \log_2 \mathfrak{q}$. One quotient realization attains all edge bounds simultaneously. Hence*

$$\Gamma_F^{\rho}(\mathcal{T}) = \mathcal{K}_F^{\rho}(\mathcal{T}) = \prod_{e \in \vec{E}(\mathcal{T})} [r_e \log_2 \mathfrak{q}, \infty)$$

under semantic traffic, with lower corner $(\lceil r_e \log_2 \mathfrak{q} \rceil)_e$ under fixed binary widths. For every nonnegative directional price vector w ,

$$\inf_P \langle w, t(P) \rangle = \sum_{e \in \vec{E}(\mathcal{T})} w_e r_e \log_2 \mathfrak{q}.$$

Proof. Fix the complete B -side input. The required B -side output ranges over an affine translate of $\text{im } M_{O_B, I_A}$, which has $\mathfrak{q}^{r_e}$ points. The directional range bound gives the converse without assuming linear encoders. Contextual equivalence identifies the sender quotient

$$\mathbb{F}_{\mathfrak{q}}^{I_A} / \ker M_{O_B, I_A},$$

of dimension r_e .

For simultaneous achievability, root the tree arbitrarily. In the inward pass, suppose child components C_i meet their parent at v and $C = \{v\} \cup_i C_i$. Since $O_{\bar{C}} \subseteq O_{\bar{C}_i}$, [Theorem 5.10](#) lets v compute Q_C from its local input and the child states. In the outward pass toward child C_j , the parent combines its local input, the inward states of all sibling components, and its incoming outward state. The target rows O_{C_j} lie outside every contributing component, so the same lemma computes the complementary state sent into C_j . Every vertex then combines its local input and incident component states to produce its placed outputs. Identifying each quotient with $\text{im } L_C$ and choosing a basis sends exactly r_e field symbols on every oriented edge in one realization. The attainable lower corner and componentwise converse give the upper orthant, already convex. $\square$

A worked three-site linear realization. Let a path 1–2–3 carry input bits x_1, x_2, x_3 at the three sites, and require outputs

$$y_1 = x_1 \oplus x_2, \quad y_2 = x_1 \oplus x_2 \oplus x_3, \quad y_3 = x_2 \oplus x_3$$

at sites 1, 2, 3, respectively. Across the left edge, the contribution of x_1 to outputs on the right has rank one, and the contribution of (x_2, x_3) to the left output also has rank one. The same is true across the right edge. The exact directional lower corner is therefore one bit in each direction on each edge. An inward/outward quotient protocol attains all four bounds simultaneously by exchanging the appropriate local parities. The example illustrates the content of the orthant theorem: interaction may change event order, but it cannot reduce the dimension of the cross-cut image, and one realization achieves all cut-rank obligations at once.

The orthant theorem determines the complete static charged-traffic geometry for this finite-field linear contract. Rounds, depth, local work, storage, approximation, randomization, variable-length expected cost, and graph-carrier coordinates define additional CRG projections and can be added without altering the exact traffic result. INDEX lies outside this linear class because its opposite input selects which distinction matters; for a linear map the cross-side image is context independent. On a graph with parallel paths, the rank converse applies to aggregate cut traffic, not independently to each arc. The one-cut transcript and rank facts are standard; the joint conclusion is the simultaneous, arbitrary-placement directed orthant even when reverse interaction is free and local encoders may be nonlinear [22, 62, 25].

5.4 Carrier graphs and network computation

A tree records nested accounting boundaries; a directed carrier graph records which local channels exist. Let $\mathcal{N} = (\mathcal{G}_{\text{car}}, \alpha_I, \alpha_O)$ be a finite directed acyclic multigraph with source-access and output-demand relations. A carrier realization assigns a finite alphabet to every arc, a local encoder from the tail's accessible sources and incoming symbols, and exact decoders at demand vertices. Its signature is the vector $\ell(P)$ of log alphabet sizes. Define the equal-instance normalized zero-error link-load region

$$\mathcal{L}_F(\mathcal{N}) = \text{cl up} \bigcup_{k \geq 1} \left\{ \frac{\ell(P)}{k} : P \text{ realizes } F^{\otimes k} \text{ exactly on } \mathcal{N} \right\}.$$

Block concatenation makes this closed upper region convex.

Theorem 5.13 (Carrier/network-code equivalence). *Under deterministic zero-error semantics on a finite acyclic carrier, single-shot carrier realizations and network function codes are identical up to internal-symbol renaming, with identical arc loads. Applying the bijection to $F^{\otimes k}$ for every k , normalizing by k , and taking closed upper block closure gives exactly the zero-error network-computation link-load region. Identity or multicast output demands specialize this to ordinary multi-source network codes under equal replication of the source instance fixed by F . This does not identify the full independent-source rate-vector region.*

Proof. Both objects assign precisely the same local encoder to each directed arc and the same decoder to each demand vertex. A topological order makes every carrier encoder a legal network-code encoder; the reverse translation changes nothing. The identity translation commutes with independent block products, normalization, and closure. $\square$

For arc capacities c_a , the asymptotic computation rate is

$$\sup (\{0\} \cup \{R > 0 : \exists \ell \in \mathcal{L}_F(\mathcal{N}), R\ell_a \leq c_a \forall a\}).$$

The adjoined zero fixes the gauge: if the carrier is zero-error unsolvable and $\mathcal{L}_F(\mathcal{N}) = \emptyset$, its computation capacity is zero. Placing carrier vertices in $\mathcal{T}$ linearly projects arc loads to tree-cut loads, but parallel arcs show the projection is not injective. Matched network-code solvability and nonlinear-code counterexamples therefore transfer exactly. Linear-versus-nonlinear comparisons require the explicit subcontract in which source and arc alphabets are vector spaces over one fixed field and all local maps are linear.

The inherited undecidability statement is specifically the variable-alphabet matched-solvability problem: given finite source, carrier, and demand incidence, decide whether some common alphabet size at least two and deterministic local encoders satisfy every demand exactly. It does not make fixed-alphabet feasibility or every network-code decision problem undecidable; a finite instance with every alphabet supplied is decidable by finite enumeration [23, 63, 64].

5.5 Temporal cut-state conservation and scoped projections

Definition 5.14 (Schedule-reachable temporal cut-state certificate). *Fix a deterministic realization with a finite acyclic event DAG, a static charged schedule, a complete site cut $A \mid B$, and a service window I . Conditional absence is an ordinary charged symbol. With receiver-local exogenous context fixed, let S_I be the receiver's retained state at the start of the window, with $\log_2 |S_I| \leq m_I$; let $\mathcal{A}_I$ contain all forward A-to-B events in the window, with $\ell_a = \log_2 |\Sigma_a|$; and let O_I be a receiver state or output required at the end, with*

$$\kappa_I = \log_2 |\text{im } O_I|$$

as compatible sender histories vary. The certificate is valid when: (i) every retained-state coordinate is available at the beginning of I ; (ii) every charged incoming event occurs before the certified output and lies on a directed path from a source-side input or state event to O_I ; (iii) every directed dependency path by which a varied source-side history can alter O_I intersects either the retained-state set or the charged incoming-event set; and (iv) each charged event is counted once under the registered accounting rule. These conditions are decidable by reachability and cut tests on the finite event DAG.

Theorem 5.15 (Cut-state conservation). *Every temporal cut-state certificate satisfies*

$$\boxed{\kappa_I \leq m_I + \sum_{a \in \mathcal{A}_I} \ell_a}, \quad \boxed{\sum_{a \in \mathcal{A}_I} \ell_a \geq (\kappa_I - m_I)_+}.$$

Interaction and nonlinear local encoding do not improve the inequality.

Proof. The initial receiver state and ordered forward transcript determine every receiver-local event, reverse message, and required output by induction over the event DAG. Their joint alphabet has at least 2^{κ_I} values and at most $2^{m_I} \prod_{a \in \mathcal{A}_I} 2^{\ell_a}$. Taking logarithms proves both forms. $\square$

Theorem 5.16 (Fractional certificate packing). *For any family of valid certificates j , with forward-event sets $\mathcal{A}_j$ and fractional weights $\alpha_j \geq 0$ satisfying*

$$\sum_{j: a \in \mathcal{A}_j} \alpha_j \leq 1 \quad \text{for every event } a,$$

total charged traffic obeys

$$\boxed{Q \geq \sum_j \alpha_j (\kappa_j - m_j)_+}.$$

Proof. This is the standard fractional-packing/linear-programming composition [112]. Multiply each instance of Theorem 5.15 by α_j , sum, exchange the order of summation, and use the per-event charging constraint. $\square$

Conservation as a realization identity. The inequality is an information balance across a temporal cut. At the beginning of the service window, the receiver possesses at most m_I bits of relevant state. During the window, the ordered forward transcript contributes at most $\sum_a \ell_a$ additional bits of log-cardinality. Every receiver state attainable at the end must be indexed by the pair consisting of initial state and forward transcript. The number of required distinctions therefore cannot exceed the product of those two state spaces. Reverse interaction may determine which forward symbol is useful, but once the receiver input is fixed it cannot create a sender-side distinction that never appears in the initial state or forward transcript.

This form makes the storage–communication tradeoff explicit: prepositioning one additional relevant bit can reduce the lower bound on forward traffic by at most one bit in the same service window. Replication, caching, and checkpointing are realizations that move along a joint state–traffic geometry.

Three classical projections follow after adding distinct service contracts.

- (i) *Bandwidth–time.* If a complete cut supplies at most B bits per unit time and no relevant state is prepositioned, cut-state conservation gives

$$BT \geq \kappa.$$

- (ii) *Atomic I/O partition.* Under the standard atomic-word red–blue pebble contract, let $H_D(S)$ be the minimum number of parts in a classical S -partition of the noninput vertices of DAG D . The parts are disjoint and exhaustive, their quotient is acyclic, each has a dominator set of at most S vertices intercepting every input-to-part path, and

$$\text{Min}(V_i) = \{v \in V_i : \text{no immediate successor of } v \text{ lies in } V_i\}$$

has cardinality at most S . Segmenting an execution after every M I/O moves creates a Hong–Kung $2M$ -partition, hence

$$Q \geq M(H_D(2M) - 1).$$

This inequality follows from the atomic-DAG partition spectrum and pebble contract; it is not obtained by silently identifying a memory move with a static tree-cut symbol.

- (iii) *Two-dimensional area–time.* Suppose every admissible balanced bisection has an orientation $A_0 \rightarrow B_0$ with forward service $B_{\text{bis}}^{A_0 \rightarrow B_0} \leq \gamma\sqrt{A}$, a valid oriented cut-state certificate requiring κ_{bis} distinctions, and at most m_{bis} relevant prepositioned bits on B_0 . With

$$\kappa_{\text{bis}}^{\text{net}} = (\kappa_{\text{bis}} - m_{\text{bis}})_+,$$

conservation yields

$$AT^2 \geq \frac{(\kappa_{\text{bis}}^{\text{net}})^2}{\gamma^2}.$$

This projection uses cut-state conservation plus the two-dimensional service and separator law.

Table 3: Temporal cut-state conservation and its separately registered projections. Bandwidth–time follows from the service-window theorem; Hong–Kung I/O and Thompson-style AT^2 require, respectively, atomic-DAG memory and when-and-where-oblivious planar-separator contracts.

Projection	Additional service contract	Resulting bound
Bandwidth–time	Complete cut, service at most B bits per unit time, no relevant prepositioned state	$BT \geq \kappa$
Atomic I/O	Red–blue pebble execution, M -move segmentation, valid $2M$ -partition	$Q \geq M(H_D(2M) - 1)$
Two-dimensional VLSI	Balanced cut, forward service at most $\gamma\sqrt{A}$, net distinction $\kappa_{\text{bis}}^{\text{net}}$	$AT^2 \geq (\kappa_{\text{bis}}^{\text{net}})^2 / \gamma^2$

The causal state-and-transcript count becomes a reusable typed certificate by registering cut, initial state, service window, and accounting rule. Fractional packing composes overlapping certificates, while the three service laws above translate the same semantic conservation statement into bandwidth–time, I/O, and area–time consequences. No genealogy may be inferred from static κ alone, and the Hong–Kung and area–time projections remain model scoped [5, 26].

6 The CRG abstraction ladder and decision-sufficient geometry

Typed units and normalization. CRG signatures are typed rather than dimensionless: state bytes, traffic bytes per operation, event counts, causal depth, work, energy, area, temperature, and reliability coordinates retain their declared units. Coordinates are not assumed commensurate. Convex combinations, norms, or scalar prices are interpreted only after the resource contract or technology ledger supplies the required normalization and unit conversion.

Definition 6.1 (Admissible decomposition system). *The decomposition system $\text{Dec}_\rho(F)$ has as objects the tree hierarchies, terminal placements, and local site rules admitted by Λ . Its structure maps are semantic hierarchy isomorphisms and connected-fiber contractions $q : \mathcal{T} \rightarrow \mathcal{T}'$ equipped with the typed pushforward $q_*\rho$ of Theorem 3.2. Identity and composition are inherited from the tree maps. A contract outside the core calculus is admissible only when its legal-realization semantics and pushforward coherence are supplied explicitly.*

Definition 6.2 (Realization bundle, signature regions, and CRG). *Fix F , ρ , and $\mathcal{T} \in \mathbf{Dec}_\rho(F)$. Let*

$$\mathcal{P}_F^\rho(\mathcal{T})$$

be the registry of ρ -legal hierarchical realizations P , modulo semantic-preserving renaming of internal states. Let

$$\sigma_{\mathcal{T}} : \mathcal{P}_F^\rho(\mathcal{T}) \longrightarrow \mathbb{R}_+^{d(\mathcal{T})}$$

extract the declared typed interface signature, or a declared projection of it. Define the exact attainable region

$$\mathcal{R}_F^\rho(\mathcal{T}) = \sigma_{\mathcal{T}}(\mathcal{P}_F^\rho(\mathcal{T})),$$

the closed upper obligation region

$$\Gamma_F^\rho(\mathcal{T}) = \text{cl}(\text{up } \mathcal{R}_F^\rho(\mathcal{T})),$$

and the convex linear-cost body

$$\mathcal{K}_F^\rho(\mathcal{T}) = \text{cl conv } \Gamma_F^\rho(\mathcal{T}).$$

The computation-realization geometry is the decomposition-indexed system

$$\mathfrak{G}_F^\rho = \left(\mathbf{Dec}_\rho(F), \left\{ \mathcal{P}_F^\rho(\mathcal{T}) \xrightarrow{\sigma_{\mathcal{T}}} \mathcal{R}_F^\rho(\mathcal{T}), \Gamma_F^\rho(\mathcal{T}), \mathcal{K}_F^\rho(\mathcal{T}) \right\}_{\mathcal{T}}, \{\pi_q\}_q \right).$$

The levels retain different information. The registry $\mathcal{P}$ distinguishes architectures, event structures, and embeddings. The exact image $\mathcal{R}$ retains jointly attainable, possibly nonconvex signatures but forgets architecture identity inside a signature fiber. The upper region Γ is the exact quotient for continuous monotone interface-obligation observations. The convex upper body $\mathcal{K}$ is the further quotient seen by additive nonnegative prices and explicitly admitted amortized mixtures. The term *geometry* denotes this resource-indexed feasible system, not a Euclidean placement, signal constellation, or smooth manifold. A selected algorithm may have an intermodule communication geometry in the classical sense [1]; here topology and technology remain unfixed until the contract and observation are chosen.

6.1 Why the geometry has four levels

The ladder is derived from the information required by the decision, not from an aesthetic preference for several sets. Three distinct losses occur.

From $\mathcal{P}$ to $\mathcal{R}$: forgetting realization identity. The signature map partitions $\mathcal{P}$ into fibers

$$\sigma_{\mathcal{T}}^{-1}(r) = \{P \in \mathcal{P}_F^\rho(\mathcal{T}) : \sigma_{\mathcal{T}}(P) = r\}.$$

Two realizations in one fiber consume the same retained coordinates, yet can have different event DAGs, schedules, layouts, reliability structure, or legal embeddings. Any objective that inspects those distinctions remains a function on $\mathcal{P}$, not merely on $\mathcal{R}$.

From $\mathcal{R}$ to Γ : forgetting exact slack. For $r \in \mathcal{R}$ and $s \geq r$ coordinatewise, s represents a provisioned budget that can serve the realization with signature r . A continuous monotone cost can never prefer s to r , so adjoining every such s changes neither the optimum nor the complete response map. Closure adds limits of attainable or provisioned sequences, exactly the information needed for continuous queries. Hence $\Gamma = \text{cl up } \mathcal{R}$ is the canonical quotient for that observation class.

From Γ to $\mathcal{K}$: forgetting nonconvex distinctions. For an additive nonnegative price w , the value of a signature is $\langle w, r \rangle$. Linear functionals have the same infimum on a set, its convex hull, and its closure. Upper closure is likewise invisible because $w \geq 0$. Therefore every such price sees precisely $\mathcal{K} = \text{cl conv } \Gamma$. A nonlinear monotone objective can still distinguish Γ from $\mathcal{K}$, and a realization-sensitive objective can distinguish $\mathcal{P}$ from all three signature images.

Definition 6.3 (Decision equivalence and sufficient representation). *Let $\mathfrak{D}$ be a class of real-valued decision objectives on a realization family. Two retained families $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}$ are $\mathfrak{D}$ -equivalent when*

$$\inf_{P \in \mathcal{A}} D(P) = \inf_{P \in \mathcal{B}} D(P) \quad \text{for every } D \in \mathfrak{D}.$$

A representation Q is sufficient for $\mathfrak{D}$ when equality of Q for two families implies their $\mathfrak{D}$ -equivalence, and complete when the converse also holds.

Theorems 6.9 and 6.11 show that Γ and $\mathcal{K}$ are complete representations for their respective decision classes. In the terminology of statistical decision theory, they play a role analogous to sufficient information structures; in the terminology of abstract interpretation, they are exact abstractions for the named query classes. CRG adds realization semantics and technology-conditioned selection to that sufficiency statement.

The diagram is also a statement about information flow through a design process. The map $\sigma_{\mathcal{T}}$ forgets which realization produced a given signature. Upper closure forgets whether a larger provisioned point is itself attained, because every monotone resource objective treats it as no better than a dominated attainable point. Convexification forgets nonconvex distinctions that no additive price can observe. No further reduction is licensed without specifying a smaller decision class.

6.2 Invariance and coherent hierarchy maps

Proposition 6.4 (Semantic isomorphism invariance). *Suppose F' is obtained from F by bijective relabelings of input alphabets, output alphabets, and terminal identities. Relabel every hierarchy and conjugate every local realization map by those bijections. This gives coordinate-preserving bijections*

$$\mathcal{R}_F^\rho(\mathcal{T}) \cong \mathcal{R}_{F'}^\rho(\mathcal{T}'), \quad \Gamma_F^\rho(\mathcal{T}) \cong \Gamma_{F'}^\rho(\mathcal{T}'), \quad \mathcal{K}_F^\rho(\mathcal{T}) \cong \mathcal{K}_{F'}^\rho(\mathcal{T}').$$

Thus the geometry depends on semantic structure and the resource contract, not on symbol names or terminal labels.

Proof. Conjugating a legal protocol or circuit by terminal bijections preserves its event structure, resource use, accounting convention, and interface traffic; the inverse relabeling recovers the original realization. $\square$

Definition 6.5 (Hierarchy contraction and signature projection). *A contraction $q : \mathcal{T} \rightarrow \mathcal{T}'$ partitions $V(\mathcal{T})$ into connected fibers and replaces each fiber by one supervertex; terminal placements are pushed forward by q . Since $\mathcal{T}$ is a tree, every edge e' of $\mathcal{T}'$ corresponds to a unique edge e of $\mathcal{T}$ joining the endpoint fibers. For an edge-indexed signature c , define*

$$(\pi_q c)_{e'} = c_e.$$

Other retained coordinates are pushed forward by their declared aggregation laws in $q_*\rho$.

Theorem 6.6 (Hierarchy-contraction consistency). *For every admissible contraction $q : \mathcal{T} \rightarrow \mathcal{T}'$:*

- (a) *the canonical directed contextual state on every surviving oriented cut is unchanged up to canonical isomorphism;*
- (b) *every legal causal realization on $\mathcal{T}$ pushes forward to a legal realization on $\mathcal{T}'$ with retained traffic vector $\pi_q c$ under $q_*\rho$; and*
- (c) *consequently,*

$$\pi_q \mathcal{R}_F^\rho(\mathcal{T}) \subseteq \mathcal{R}_F^{q_*\rho}(\mathcal{T}'), \quad \pi_q \Gamma_F^\rho(\mathcal{T}) \subseteq \Gamma_F^{q_*\rho}(\mathcal{T}'), \quad \pi_q \mathcal{K}_F^\rho(\mathcal{T}) \subseteq \mathcal{K}_F^{q_*\rho}(\mathcal{T}').$$

Proof. For a surviving edge, contraction does not change the two terminal sets separated by its cut, so the contextual equivalence relation is identical. For a general protocol, treat each connected fiber as one local superprocessor and execute all events internal to it locally. Only messages crossing the unique fine edge corresponding to a coarse edge remain, producing $\pi_q c$. Upper closure, topological closure, convexification, and coordinate projection give the stated inclusions. $\square$

Theorem 6.7 (Decomposition-system coherence). *The structure maps satisfy*

$$\pi_{\text{id}_{\mathcal{T}}} = \text{id}, \quad \pi_{r \circ q} = \pi_r \circ \pi_q,$$

and their contracts satisfy $(r \circ q)_*\rho = r_*(q_*\rho)$. Semantic hierarchy isomorphisms act by coordinate bijections and commute with the projections. Thus $\mathfrak{S}_F^\rho$ is a coherent decomposition-indexed system, not a bare collection of feasible regions.

Proof. Every surviving coarse edge corresponds to the same unique original tree edge whether contractions are applied at once or successively, so coordinate restriction composes exactly. Connected-fiber localization and aggregation of retained resource charges are associative by Theorem 3.3. Relabelings only rename coordinates and local symbols and hence commute with projection. $\square$

6.3 Monotone reconstruction and linear duality

Definition 6.8 (Monotone interface-cost response). *For a continuous coordinatewise nondecreasing $\Phi : \mathbb{R}_+^{d(\mathcal{T})} \rightarrow \mathbb{R}_+$, define*

$$V_{F,\rho,\mathcal{T}}(\Phi) = \inf_{r \in \mathcal{R}_F^\rho(\mathcal{T})} \Phi(r),$$

with value $+\infty$ if the hierarchy is infeasible.

Theorem 6.9 (Reconstruction by monotone interface observations). *For every feasible hierarchy $\mathcal{T}$:*

(a) *every continuous coordinatewise nondecreasing cost satisfies*

$$V_{F,\rho,\mathcal{T}}(\Phi) = \inf_{x \in \Gamma_F^\rho(\mathcal{T})} \Phi(x);$$

(b) *the full response $\Phi \mapsto V_{F,\rho,\mathcal{T}}(\Phi)$ reconstructs $\Gamma_F^\rho(\mathcal{T})$ exactly. In particular, for $a \in \mathbb{R}_+^{d(\mathcal{T})}$, let*

$$\Phi_a(x) = \|(x - a)_+\|_\infty.$$

Then

$$a \in \Gamma_F^\rho(\mathcal{T}) \iff V_{F,\rho,\mathcal{T}}(\Phi_a) = 0;$$

(c) *the decomposition-indexed response functional $(\mathcal{T}, \Phi) \mapsto V_{F,\rho,\mathcal{T}}(\Phi)$ and the family of upper obligation regions $\{\Gamma_F^\rho(\mathcal{T})\}_\mathcal{T}$ determine one another, up to coordinate relabelings declared by ρ .*

Conversely, those regions determine every query in this observation class. The completeness claim does not extend automatically to objectives depending on a realization fiber, causal timing, switching activity, or physical embedding.

Proof. Monotonicity gives $\inf_{\text{up}} \mathcal{R} \Phi = \inf_{\mathcal{R}} \Phi$: increasing any provisioned coordinate cannot improve the cost. Continuity preserves the infimum under closure, proving (a).

If $a \in \Gamma$, then $\Phi_a(a) = 0$. Conversely, if the infimum is zero, choose $x_n \in \Gamma$ with $\|(x_n - a)_+\|_\infty \rightarrow 0$ and set $y_n = x_n \vee a$ coordinatewise. Upper closure gives $y_n \in \Gamma$, while $y_n \rightarrow a$; closedness gives $a \in \Gamma$. Applying this membership test on every admissible hierarchy proves (c), and the converse follows from (a). $\square$

Definition 6.10 (Lower linear support cost). *For $w \in \mathbb{R}_+^{d(\mathcal{T})}$, define*

$$h_{F,\rho,\mathcal{T}}(w) = \inf_{c \in \mathcal{K}_F^\rho(\mathcal{T})} \langle w, c \rangle.$$

Theorem 6.11 (Linear-cost-body duality). *Let $\mathcal{K} \subseteq \mathbb{R}_+^m$ be nonempty, closed, convex, and upper closed, and put $h_{\mathcal{K}}(w) = \inf_{x \in \mathcal{K}} \langle w, x \rangle$ for $w \geq 0$. Then*

$$\mathcal{K} = \{x \in \mathbb{R}_+^m : \langle w, x \rangle \geq h_{\mathcal{K}}(w) \text{ for every } w \in \mathbb{R}_+^m\}.$$

Thus the optimum under every additive nonnegative interface-price system uniquely determines the linear-cost body [2].

Proof. Every $x \in \mathcal{K}$ satisfies the inequalities by definition. Conversely, let $x \notin \mathcal{K}$. Strong separation gives a and α with

$$\langle a, x \rangle < \alpha \leq \langle a, y \rangle \quad \text{for every } y \in \mathcal{K}.$$

The separator must satisfy $a \geq 0$: if $a_j < 0$, upper closure would put $y + te_j$ in $\mathcal{K}$ for every $t \geq 0$, driving $\langle a, y + te_j \rangle$ to $-\infty$. Hence a is an allowed price vector and $\langle a, x \rangle < h_{\mathcal{K}}(a)$. $\square$

Corollary 6.12 (Linear-price completeness). *For a fixed hierarchy, $\mathcal{K}_F^\rho(\mathcal{T})$ and the response $h(w)$ determine one another, and distinct bodies are separated by a nonnegative price vector. Convexification is exact for additive linear interface prices; general monotone costs require Γ , while objectives distinguishing realizations with one signature require $\mathcal{P}$.*

Definition 6.13 (Decision-response equivalence). *Let $\mathfrak{C}$ be a class of objectives on a common signature space. Two nonempty realization families $\mathcal{A}$ and $\mathcal{B}$ are $\mathfrak{C}$ -equivalent when*

$$\inf_{P \in \mathcal{A}} C(\sigma_{\mathcal{T}}(P)) = \inf_{P \in \mathcal{B}} C(\sigma_{\mathcal{T}}(P)) \quad \text{for every } C \in \mathfrak{C}.$$

A statistic of a realization family is decision sufficient for $\mathfrak{C}$ when equality of that statistic is equivalent to $\mathfrak{C}$ -equivalence.

Theorem 6.14 (Canonical decision quotients). *Fix F , ρ , and $\mathcal{T}$.*

- (a) *For the class $\mathfrak{C}_\uparrow$ of continuous coordinatewise nondecreasing signature objectives, the map $\mathcal{A} \mapsto \Gamma_{\mathcal{A}}$ is a complete decision-sufficient statistic.*
- (b) *For the class $\mathfrak{C}_+$ of additive objectives $c \mapsto \langle w, c \rangle$ with $w \geq 0$, the map $\mathcal{A} \mapsto \mathcal{K}_{\mathcal{A}}$ is a complete decision-sufficient statistic.*
- (c) *Neither quotient is complete for objectives that distinguish realizations in the same signature fiber; those objectives require the registry $\mathcal{P}$ or an enriched signature.*

Consequently, the ladder $\mathcal{P} \rightarrow \mathcal{R} \rightarrow \Gamma \rightarrow \mathcal{K}$ is an ordered family of exact observation quotients rather than a sequence of heuristic summaries.

Proof. Part (a) follows from Theorem 6.9: the complete monotone response reconstructs the upper region, and the upper region computes every response. Part (b) follows from Theorems 6.11 and 6.12. For (c), choose two distinct realizations P_1, P_2 with $\sigma_{\mathcal{T}}(P_1) = \sigma_{\mathcal{T}}(P_2)$ and an objective that assigns different costs to their schedules, causal structures, or embeddings. Every statistic factoring only through the common signature gives them the same value, whereas the realization-sensitive objective separates them. $\square$

The theorem is a deterministic design analogue of statistical decision sufficiency: the sufficient object is defined by the family of questions that must remain answerable. It also supplies the precision order for abstraction. When only additive prices remain, $\mathcal{K}$ is sufficient. When nonlinear monotone constraints or bottleneck objectives remain, Γ is required. When exact discrete alternatives matter, $\mathcal{R}$ is required. When schedule, architecture, or physical embedding matters, the realization fiber must be retained.

6.4 Safe commitment and insufficiency of scalar summaries

For a retained family $\emptyset \neq \mathcal{A} \subseteq \mathcal{P}_F^\rho(\mathcal{T})$ and an admissible signature objective C , define the commitment regret

$$\text{Regret}_C(\mathcal{A}) = \inf_{P \in \mathcal{A}} C(\sigma_{\mathcal{T}}(P)) - \inf_{P \in \mathcal{P}_F^\rho(\mathcal{T})} C(\sigma_{\mathcal{T}}(P)) \geq 0. \quad (6.1)$$

A reduction is safe for a declared decision class exactly when it leaves every optimum in that class unchanged. The next theorem identifies the corresponding geometric invariants.

Theorem 6.15 (Safe-Commitment Law: universal lossless retention). *Let $\emptyset \neq \mathcal{A} \subseteq \mathcal{P}_F^\rho(\mathcal{T})$ be a retained realization family and define*

$$\Gamma_{\mathcal{A}} = \text{clup } \sigma_{\mathcal{T}}(\mathcal{A}), \quad \mathcal{K}_{\mathcal{A}} = \text{clconv } \Gamma_{\mathcal{A}}.$$

Then:

- (i) *$\mathcal{A}$ preserves the optimum of every continuous coordinatewise nondecreasing signature objective if and only if $\Gamma_{\mathcal{A}} = \Gamma_F^\rho(\mathcal{T})$;*
- (ii) *it preserves the optimum under every additive nonnegative price if and only if $\mathcal{K}_{\mathcal{A}} = \mathcal{K}_F^\rho(\mathcal{T})$; and*
- (iii) *if either equality fails, a member of the corresponding observation class has strictly positive commitment regret in the sense of Equation (6.1).*

Proof. Suppose first that $\Gamma_{\mathcal{A}} = \Gamma_F^\rho(\mathcal{T})$. Every continuous monotone objective has the same infimum on a realization family and on the upper closure of its signatures by Theorem 6.9; hence the retained and full optima agree. The same argument with linear-cost duality proves sufficiency of $\mathcal{K}_{\mathcal{A}} = \mathcal{K}_F^\rho(\mathcal{T})$ for additive prices.

For necessity in the monotone class, assume every monotone optimum agrees. For any $a \in \mathbb{R}_+^{d(\mathcal{T})}$, apply the membership probe $\Phi_a(x) = \|(x - a)_+\|_\infty$. The retained response is zero exactly when $a \in \Gamma_{\mathcal{A}}$, and the full response is zero exactly when $a \in \Gamma_F^\rho(\mathcal{T})$. Equality of all responses therefore gives pointwise equality of the two regions.

For necessity in the linear-price class, equality of every nonnegative linear response gives the same support function on the nonnegative orthant. Theorem 6.11 reconstructs the two closed convex upper bodies from those responses, so they are equal.

Finally, if $\Gamma_{\mathcal{A}} \neq \Gamma_F^\rho(\mathcal{T})$, choose a in their set difference. The probe Φ_a has full optimum zero and retained optimum strictly positive, producing commitment regret. If the convex bodies differ, strong separation by a nonnegative vector w gives an additive price whose retained optimum is strictly larger than the full optimum. These objectives are explicit witnesses to the information lost by the reduction. $\square$

Corollary 6.16 (Strict engineering-cost separation). *Let $\Gamma_{\mathcal{A}} \subsetneq \Gamma$ be closed upper sets generated by a retained family $\mathcal{A}$ and by the full legal realization family, respectively. Then there exists a continuous, coordinatewise strictly increasing, coercive cost $\Phi : \mathbb{R}_+^d \rightarrow \mathbb{R}_+$ such that*

$$\inf_{x \in \Gamma} \Phi(x) < \inf_{x \in \Gamma_{\mathcal{A}}} \Phi(x).$$

Consequently, loss of the monotone-decision geometry is witnessed by an engineering-style cost that penalizes every declared resource, not only by a membership probe.

Proof. Choose $a \in \Gamma \setminus \Gamma_{\mathcal{A}}$ and set

$$\delta := \inf_{x \in \Gamma_{\mathcal{A}}} \|(x - a)_+\|_{\infty}.$$

Closed upperness implies $\delta > 0$: otherwise a sequence in $\Gamma_{\mathcal{A}}$ would approach a after taking coordinatewise maxima with a , and closedness would place a in $\Gamma_{\mathcal{A}}$. Choose

$$0 < \varepsilon < \frac{\delta}{1 + \sum_i a_i}$$

and define

$$\Phi(x) := \|(x - a)_+\|_{\infty} + \varepsilon \sum_i x_i.$$

The positive linear term makes Φ coordinatewise strictly increasing and coercive. Every $x \in \Gamma_{\mathcal{A}}$ satisfies $\Phi(x) \geq \delta$, whereas $\Phi(a) = \varepsilon \sum_i a_i < \delta$. Since $a \in \Gamma$, strict separation follows. $\square$

Corollary 6.17 (Finite-registry retention tests). *Suppose $\mathcal{P}_F^{\rho}(\mathcal{T})$ is finite.*

- (a) *A retained family is complete for all continuous monotone signature objectives exactly when every discarded signature is coordinatewise dominated by a retained signature.*
- (b) *It is complete for all additive nonnegative prices exactly when every full signature lies in $\text{conv}(\sigma_{\mathcal{T}}(\mathcal{A})) + \mathbb{R}_+^{d(\mathcal{T})}$.*
- (c) *If a named finite technology ledger Θ has attained minima, the smallest requirement is argmin hitting: $\mathcal{A} \cap \text{argmin}_P C_{\theta}(P) \neq \emptyset$ for every $\theta \in \Theta$.*

Proof. For a finite set, the minimal points of its upper closure are precisely the nondominated signatures, proving (a). Convexification of a finite upper set is $\text{conv}(\sigma(\mathcal{P})) + \mathbb{R}_+^d$, proving (b). Part (c) is the finite-ledger specialization proved in [Theorem 16.10](#). $\square$

Safe-commitment principle. Within the declared observation class, a realization may be discarded only when the relevant geometry is unchanged. Equivalently, discard a realization only when every future decision in that class is unable to detect that it is gone.

Derivation of the law

The equivalence can be read constructively in both directions.

Sufficiency. If $\Gamma_{\mathcal{A}} = \Gamma$, then every continuous monotone objective has the same infimum on the retained and full families because both optimize over the same complete upper region. If $\mathcal{K}_{\mathcal{A}} = \mathcal{K}$, the same statement holds for every additive nonnegative price by linear-cost duality.

Necessity. Suppose the retained upper region omits a point $a \in \Gamma$. The probe

$$\Phi_a(x) = \|(x - a)_+\|_{\infty}$$

has full-family optimum zero but retained-family optimum strictly positive; it is therefore an explicit witness of commitment regret. If instead the convex bodies differ, a nonnegative separating price w gives a strict value gap. The law is thus not a heuristic preference for diversity. It is a complete equivalence between a geometric equality and preservation of an entire class of future decisions.

Three worked retention regimes

Universal monotone regime.

Every resource objective that is continuous and worsens when any coordinate increases remains admissible. A discarded finite-registry point is safe exactly when a retained point dominates it.

Additive-price regime.

Future decisions are restricted to $w \cdot r$ with $w \geq 0$. A discarded point can remain safe even when it is not dominated, provided it lies in the convex upper hull generated by retained points.

Named technology ledger.

Only a finite set of technology/objective pairs remains open. It is sufficient to retain a family that intersects the argmin set of every ledger entry, as formalized later by the argmin-hitting theorem. This quotient can be much smaller than either universal geometry.

These regimes separate a universal theorem from a practical retention policy. The decision class is declared first; the smallest sufficient realization record follows from that declaration.

For a finite registry, universal monotone retention means that every discarded signature is dominated by a retained one. Linear-price retention permits the weaker test

$$\mathcal{R}_F^\rho(\mathcal{T}) \subseteq \text{conv}(\sigma_{\mathcal{T}}(\mathcal{A})) + \mathbb{R}_+^{d(\mathcal{T})}.$$

These statements are universal only relative to their named observation classes; a fixed finite technology ledger can require less data.

Theorem 6.18 (Strict insufficiency of marginal and scalar summaries). *There are two closed upper feasible regions with identical coordinate floors, minimum total traffic, and minimum bottleneck width, but different technology-price responses. Let*

$$\Gamma_A = \text{up}\{(1, 2), (2, 1)\}, \quad \Gamma_B = \text{up}\{(1, 3), (2, 1)\}.$$

Both satisfy

$$(\inf x_1, \inf x_2) = (1, 1), \quad \inf(x_1 + x_2) = 3, \quad \inf \max\{x_1, x_2\} = 2.$$

Yet for $w = (10, 1)$,

$$\inf_{x \in \Gamma_A} \langle w, x \rangle = 12, \quad \inf_{x \in \Gamma_B} \langle w, x \rangle = 13.$$

The jointly realizable region therefore retains operational information lost by coordinatewise minima, total width, and bottleneck width.

Proof. Evaluate the two Pareto-minimal points of each upper region. Under $w = (10, 1)$, $(1, 2)$ minimizes Γ_A at 12, while $(1, 3)$ minimizes Γ_B at 13; $(2, 1)$ costs 21 in both. $\square$

Example 6.19 (Why the exact region cannot be convexified universally). *Suppose the only exact signatures are $r = (0, 2)$ and $s = (2, 0)$. Their convex linear-cost body contains $(1, 1)$. Every linear price has the same infimum on the two raw points and on the body, but the monotone convex objective $\Phi(c) = \max\{c_1, c_2\}$ has exact optimum 2 and convex-relaxation optimum 1. Thus $\mathcal{K}$ is exact for linear prices, $\mathcal{R}$ preserves discrete joint alternatives, and $\mathcal{P}$ is required to preserve architecture identity within each signature fiber.*

Proposition 6.20 (Price reversal for incomparable signatures). *If $r, s \in \mathbb{R}_+^m$ are incomparable in coordinate order, there are nonnegative prices w, w' such that*

$$\langle w, r \rangle < \langle w, s \rangle, \quad \langle w', s \rangle < \langle w', r \rangle.$$

No price-independent monotone scalar ranking can therefore agree with every nonnegative linear price system.

Proof. Incomparability gives coordinates i, j with $r_i < s_i$ and $r_j > s_j$. Take $w = e_i$ and $w' = e_j$. $\square$

7 Monotonicity and composition

Proposition 7.1 (Resource monotonicity). *If $\rho \preceq \rho'$, then on every hierarchy admissible under both contracts,*

$$\mathcal{R}_F^\rho(\mathcal{T}) \subseteq \mathcal{R}_F^{\rho'}(\mathcal{T}), \quad \Gamma_F^\rho(\mathcal{T}) \subseteq \Gamma_F^{\rho'}(\mathcal{T}), \quad \mathcal{K}_F^\rho(\mathcal{T}) \subseteq \mathcal{K}_F^{\rho'}(\mathcal{T}),$$

and

$$h_{F, \rho', \mathcal{T}}(w) \leq h_{F, \rho, \mathcal{T}}(w) \quad \text{for all } w \geq 0.$$

If Λ' also admits more hierarchies than Λ , a global hierarchy optimum can only improve further.

Proof. Relaxation is inclusion of legal realizations by definition. Applying $\sigma_{\mathcal{T}}$, upper closure, closure, and convexification preserves inclusion. Taking an infimum over the larger body proves the price inequality. $\square$

Proposition 7.2 (Local data processing). *Suppose $F' = p_{\text{out}} \circ F \circ p_{\text{in}}$, where p_{in} and p_{out} are implementable locally at the input and output terminals for free under ρ . Every realization of F then induces a realization of F' with the same interface signature. Hence*

$$\mathcal{K}_{F'}^{\rho}(\mathcal{T}) \supseteq \mathcal{K}_F^{\rho}(\mathcal{T}), \quad h_{F',\rho,\mathcal{T}}(w) \leq h_{F,\rho,\mathcal{T}}(w).$$

Local processing can encode or discard communication demand, but cannot make the derived task more expensive under the same interface prices.

Proof. Precompose each input terminal with p_{in} , run the realization of F , and postcompose its output terminals with p_{out} . The added maps are local and free, so no interface coordinate changes. Region inclusion and the support-cost inequality follow. $\square$

Theorem 7.3 (Exact product additivity for canonical summaries). *Let F_1 and F_2 have disjoint input and output alphabets placed on the same tree, and consider their product computation $F_1 \times F_2$. In the separable directed-summary model,*

$$Q_{u \rightarrow v}^{F_1 \times F_2} \cong Q_{u \rightarrow v}^{F_1} \times Q_{u \rightarrow v}^{F_2}$$

on every oriented edge, and therefore

$$\kappa_{u \rightarrow v}(F_1 \times F_2) = \kappa_{u \rightarrow v}(F_1) + \kappa_{u \rightarrow v}(F_2).$$

If $\hat{\kappa} = \lceil \kappa \rceil$ is the minimum fixed-bit width of one canonical summary, then

$$\hat{\kappa}(F_1) + \hat{\kappa}(F_2) - 1 \leq \hat{\kappa}(F_1 \times F_2) \leq \hat{\kappa}(F_1) + \hat{\kappa}(F_2).$$

Semantic capacities add exactly; joint fixed-bit packing can save at most one bit per directed interface relative to separate rounding.

Proof. If two product-side assignments are equivalent for both component outputs on the opposite side, they are equivalent for the ordered output pair. Conversely, if they are inequivalent for either component, fix all variables of the other component and use a distinguishing opposite-side context for the first; the ordered output differs. The equivalence relation is therefore the Cartesian product relation, class cardinalities multiply, and logarithms add. The fixed-bit inequalities are the elementary bounds for the ceiling of a sum. $\square$

Proposition 7.4 (Constructive subadditivity and communication synergy). *Let F_1, F_2 be governed by ρ_1, ρ_2 , and suppose the product contract $\rho_1 \oplus \rho_2$ permits side-by-side execution with additive local resources. In any interactive realization model closed under that parallel execution,*

$$h_{F_1 \times F_2, \rho_1 \oplus \rho_2, \mathcal{T}}(w) \leq h_{F_1, \rho_1, \mathcal{T}}(w) + h_{F_2, \rho_2, \mathcal{T}}(w).$$

Define

$$\text{Syn}_w(F_1, F_2) = h_{F_1, \rho_1, \mathcal{T}}(w) + h_{F_2, \rho_2, \mathcal{T}}(w) - h_{F_1 \times F_2, \rho_1 \oplus \rho_2, \mathcal{T}}(w) \geq 0.$$

This is the price-weighted communication saved by shared encodings, intermediate computation, or control relative to separate realization. Exact direct-sum behavior is the case $\text{Syn}_w = 0$; information-complexity direct-sum results are neighboring tools [3, 4].

Proof. Choose realizations of the two tasks arbitrarily close to their separate infima and execute them side by side. The product contract makes their signatures additive, so the resulting price is the sum. Taking the infimum over all joint realizations gives the inequality and nonnegativity of Syn_w . $\square$

Serial composition has the analogous constructive bound: concatenate realizations and eliminate the intermediate terminal. Joint optimization may improve it by fusing the intermediate representation; that serial fusion gain is the sequential analogue of Syn_w . It is a quantity defined by the full geometry, not an automatic equality.

Part II

Scaling laws from resource geometry

8 A scale profile from one legal realization

Fix a computation F , a resource contract ρ , an admissible recursive hierarchy $\mathcal{T}$, and one ρ -legal realization P . The foundations associate a typed interface signature with every boundary of $\mathcal{T}$ and collect the jointly attainable signatures in a realization region. A scale profile is a projection of one such jointly attainable signature; it is not a sequence obtained by minimizing each scale independently.

Let G denote total local work in the declared accounting unit. At hierarchy level k , let $\Pi_{\text{part},k}$ be the module partition and let $g_k = G/|\Pi_{\text{part},k}|$ be its nominal scale. The level is ζ -size-regular when every nonresidual module $M \in \Pi_{\text{part},k}$ satisfies

$$\zeta g_k \leq |M| \leq g_k, \quad 0 < \zeta \leq 1,$$

with the same ζ across the measured levels. Fix a typed projection π_χ of the interface signature—for example, fixed-bit terminal incidence per operation under a declared fanout convention—and write

$$T(M) = \pi_\chi([\sigma_{\mathcal{T}}(P)]_M) > 0$$

on the levels retained for logarithmic analysis.

Definition 8.1 (Fixed-realization level-mean profile). *For a size-regular level, define*

$$R_k = \frac{1}{|\Pi_{\text{part},k}|} \sum_{M \in \Pi_{\text{part},k}} T(M), \quad g_k = \frac{G}{|\Pi_{\text{part},k}|}.$$

The sequence $\{(g_k, R_k)\}$ is the level-mean scale profile of $(F, \rho, P, \mathcal{T}, \pi_\chi)$. Between adjacent hierarchy scales, its canonical interpolation is log-linear: $\log R$ is affine in $\log g$. Every derivative or local slope below refers to that interpolation unless a smooth continuum model is explicitly supplied.

For irregular partitions, either bin modules into fixed-ratio size classes or retain the full per-module scatter. The nominal value $G/|\Pi_{\text{part},k}|$ is not by itself a valid scale label when module sizes vary without control. Likewise, profiles with different terminal placement, hierarchy, traffic type, or accounting rule belong to different projections and must not be silently pooled.

The absolute terminal incidence at a level is

$$C(g_k) = \sum_{M \in \Pi_{\text{part},k}} T(M).$$

For $|\Pi_{\text{part},k}| \asymp G/g_k$,

$$R(g_k) = \frac{g_k}{G} C(g_k)$$

up to the fixed size-regularity constants. Circuit cut profiles, memory-traffic hierarchies, and component-plus-interface spectra are thus different typed projections of the same kind of fixed-realization object. The obligation region $\Gamma_F^\rho(\mathcal{T})$ may supply outer bounds on such profiles, but the profile itself is always traced through one legal realization across all measured levels. This coherence condition is fundamental: independently minimizing each scale would assemble a curve whose points may come from mutually incompatible algorithms, schedules, or hierarchies.

From incidence to a Rent-like profile. Suppose a level has m_k approximately equal modules, each of size $g_k \asymp G/m_k$, and aggregate external incidence $C(g_k)$. The mean incidence is

$$R(g_k) = \frac{C(g_k)}{m_k} \asymp \frac{g_k}{G} C(g_k).$$

If the aggregate incidence scales as $C(g) \asymp G g^{p-1} \ell(g)$, then the level mean scales as $R(g) \asymp g^p \ell(g)$. Thus the familiar pin-versus-block exponent is a projection of how aggregate boundary incidence changes along one hierarchy. The factor g/G is not cosmetic: it converts a level total into a per-module quantity and makes the measurement convention explicit.

9 Stable dilation limits and finite hierarchy phase

Definition 9.1 (Regular variation). *A measurable positive function R is regularly varying with index p , written $R \in \text{RV}_p$, if*

$$\lim_{g \rightarrow \infty} \frac{R(\lambda g)}{R(g)} = \lambda^p \quad \text{for every } \lambda > 0.$$

A function in RV_0 is slowly varying.

Theorem 9.2 (Stable-scale limit). *Let $R : (g_0, \infty) \rightarrow (0, \infty)$ be measurable. Suppose that for every $\lambda > 0$ the finite positive limit*

$$Q(\lambda) = \lim_{g \rightarrow \infty} \frac{R(\lambda g)}{R(g)}$$

exists. Then there is a unique $p \in \mathbb{R}$ such that

$$Q(\lambda) = \lambda^p$$

for all $\lambda > 0$. Consequently,

$$\boxed{R(g) = g^p \ell(g)}$$

for a slowly varying function ℓ .

Proof. For $\lambda, \mu > 0$,

$$\frac{R(\lambda \mu g)}{R(g)} = \frac{R(\lambda \mu g)}{R(\mu g)} \frac{R(\mu g)}{R(g)}.$$

Taking limits gives $Q(\lambda \mu) = Q(\lambda)Q(\mu)$. The pointwise limit of measurable functions is measurable, so Q is a positive measurable multiplicative function. Setting $\varphi(t) = \log Q(e^t)$ gives a measurable additive function and hence $\varphi(t) = pt$ for a unique p . Therefore $Q(\lambda) = \lambda^p$. Defining $\ell(g) = R(g)/g^p$ gives $\ell(\lambda g)/\ell(g) \rightarrow 1$ [54]. $\square$

Corollary 9.3 (Pure power as an exact fixed point). *If $R(\lambda g) = \lambda^p R(g)$ for every $g, \lambda > 0$, then*

$$R(g) = t g^p$$

for $t = R(1)$. Conversely, every pure power has exact scale covariance.

The exact power is therefore one fixed point. Stable dilation limits also admit a slowly varying envelope, which can remain numerically important over every realizable design scale. The theorem separates two pieces of the profile:

$$\log R(g) = p \log g + \log \ell(g).$$

The first term is the scale index; the second is the finite-scale envelope. A fitted line estimates both at once. CRG therefore treats a measured exponent as a windowed observation of a typed realization profile rather than as an intrinsic scalar attached to the computation.

9.1 Preferred dilation factors

Digital hierarchies often repeat only at a declared factor $b > 1$. Covariance under one factor is weaker than regular variation.

Theorem 9.4 (Discrete scale covariance). *Fix $b > 1$. A positive function satisfies*

$$R(bg) = b^p R(g) \quad \text{for all } g > 0$$

if and only if

$$\boxed{R(g) = g^p \psi(\log_b g)}$$

for a positive 1-periodic function ψ .

Proof. Define $\psi(t) = b^{-pt} R(b^t)$. The covariance relation gives $\psi(t+1) = \psi(t)$. Substitution proves the converse. $\square$

This is the discrete-scale-invariance mechanism behind log-periodic corrections [55]. A binary or quadtree hierarchy can have a stable mean exponent while retaining a phase oscillation in logarithmic coordinates.

Theorem 9.5 (Two-scale criterion). *Let R be eventually nondecreasing. Suppose there exist $a, b > 1$ and $p \geq 0$ such that $\log a / \log b$ is irrational and*

$$\frac{R(ag)}{R(g)} \rightarrow a^p, \quad \frac{R(bg)}{R(g)} \rightarrow b^p.$$

Then $R \in \text{RV}_p$.

Proof. The subgroup $\{a^m b^n : m, n \in \mathbb{Z}\}$ is dense in $\mathbb{R}_+$. Repeated application of the two limits, including reciprocal limits for negative powers, gives $R(hg)/R(g) \rightarrow h^p$ for every subgroup element h . For arbitrary $\lambda > 0$, choose subgroup elements $h_- < \lambda < h_+$ arbitrarily close to λ . Eventual monotonicity gives

$$\frac{R(h_-g)}{R(g)} \leq \frac{R(\lambda g)}{R(g)} \leq \frac{R(h_+g)}{R(g)}.$$

First take $g \rightarrow \infty$ and then $h_{\pm} \rightarrow \lambda$. □

One preferred factor can therefore preserve hierarchy phase; stable scaling at two incommensurate factors removes that ambiguity. In physical terms, a binary hierarchy can revisit the same structural phase every time scale is multiplied by two. A second incommensurate dilation probes the profile at a dense set of logarithmic phases and forces the periodic factor to collapse to a constant.

9.2 Normalized dilation action and phase ratios

Definition 9.6 (Normalized dilation action). *For a positive measurable profile, considered modulo a positive amplitude, choose the representative $R(1) = 1$ and define*

$$(\mathcal{D}_a R)(x) = \frac{R(ax)}{R(a)}, \quad a > 0.$$

Then $\mathcal{D}_a \mathcal{D}_c = \mathcal{D}_{ac}$.

Proposition 9.7 (Fixed points and scale cycles). *A positive measurable profile is fixed by every $\mathcal{D}_a$ exactly when $R(x) = x^p$. For a declared hierarchy factor $b > 1$, it is fixed by $\mathcal{D}_b^h$ exactly when*

$$R(x) = x^p \psi(\log_b x)$$

for a positive h -periodic ψ normalized by $\psi(0) = 1$. Finally, $\mathcal{D}_a R \rightarrow x^p$ locally uniformly as $a \rightarrow \infty$ exactly when $R \in \text{RV}_p$.

Proof. Full invariance is the measurable multiplicative equation $R(ax) = R(a)R(x)$, whose positive solutions are powers. Discrete invariance follows by setting $\psi(t) = b^{-pt} R(b^t)$ and imposing h -periodicity; the converse is immediate. The last statement is the uniform convergence theorem for regular variation [54]. □

The following result characterizes the observable scalar sequence. It does not infer the latent mechanism that generated that sequence.

Theorem 9.8 (Phase-ratio characterization). *Fix $g_k = g_0 b^k$ and a positive sequence Y_k with its canonical log-linear interpolation. For an integer $h \geq 1$, the phasewise limits*

$$\frac{Y_{nh+j+1}}{Y_{nh+j}} \rightarrow c_j > 0, \quad j = 0, \dots, h-1,$$

exist if and only if

$$\boxed{Y(g) = Y(g_0)(g/g_0)^p L(g) \psi(\log_b(g/g_0))},$$

where

$$p = \frac{\log \prod_{j=0}^{h-1} c_j}{h \log b},$$

$L \in \text{RV}_0$, and ψ is positive, log-linearly interpolated, and h -periodic. Under this interpolation, $Y \in \text{RV}_p$ exactly when all $c_j = b^p$. If

$$\sum_k \left| \log \frac{Y_{k+1}/Y_k}{c_{k \bmod h}} \right| < \infty,$$

then $L(g) \rightarrow L_\infty \in (0, \infty)$.

Proof. Set $a_0 = 1$ and $a_{j+1} = a_j c_j b^{-p}$; the definition of p gives $a_h = a_0$. For $k = nh + j$, define

$$L_k = \frac{Y_k}{Y_0 b^{pk} a_j}.$$

Then $L_{k+1}/L_k \rightarrow 1$. An interval from g to λg crosses only a bounded number of hierarchy cells, and the logarithmic increment on each late cell tends to zero. Hence the log-linear interpolation of L_k is slowly varying. Periodically interpolate a_j to obtain ψ . Substitution proves the reverse implication. A nonconstant ψ leaves distinct phase-subsequence ratios, whereas absolute summability makes $\log L_k$ a convergent series. $\square$

The adjacent log slopes

$$s_k = \frac{\log(Y_{k+1}/Y_k)}{\log b}$$

therefore converge to one value in the regularly varying class, converge to an h -cycle in the finite-phase class, and do neither outside those output classes. The theorem provides an observable classification independent of a latent mechanism. It answers the empirical question first—what ratio law does the measured profile exhibit?—and leaves the mechanistic question to the positive-mode and mixture theorems that follow.

10 Windowed estimators and finite-scale bias

For a differentiable profile define the local logarithmic slope

$$p_{\text{loc}}(g) = \frac{d \log R(g)}{d \log g}.$$

If $R(g) = g^p \ell(g)$, then

$$p_{\text{loc}}(g) = p + \frac{d \log \ell(g)}{d \log g}.$$

The correction need not be monotone and need not be small on realizable scales.

Theorem 10.1 (Multiplicative-window convergence). *Let $R(g) = g^p \ell(g)$ with ℓ slowly varying. Fix distinct multipliers $\lambda_1, \dots, \lambda_J > 0$. At anchor t , fit the unweighted ordinary-least-squares slope of*

$$(\log(t\lambda_j), \log R(t\lambda_j))_{j=1}^J.$$

Then

$$\boxed{\widehat{p}(t) \rightarrow p.}$$

Proof. Let a_j be the OLS slope weights. They are independent of t and satisfy

$$\sum_j a_j = 0, \quad \sum_j a_j \log \lambda_j = 1.$$

Therefore

$$\widehat{p}(t) = p + \sum_j a_j \log \ell(t\lambda_j) = p + \sum_j a_j \log \frac{\ell(t\lambda_j)}{\ell(t)}.$$

Every term tends to zero by slow variation. $\square$

A fitted slope can thus converge under outward shifts of a fixed multiplicative window even when the exact relation is not a pure power. The slowly varying term controls finite-window bias. For example, $R(g) \asymp g/\log g$ belongs to RV_1 but has smooth local slope

$$1 - \frac{1}{\log g},$$

which converges slowly. Butterfly and FFT-like recursive structures provide canonical examples of this correction [56].

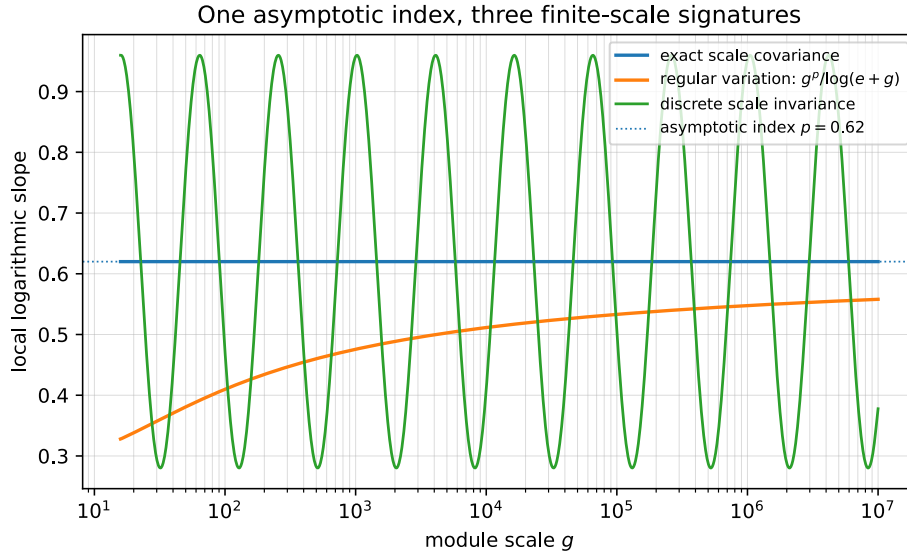


Figure 2: Exact scale covariance, regular variation with a logarithmic correction, and discrete scale invariance can share an asymptotic index while producing different finite-scale slopes.

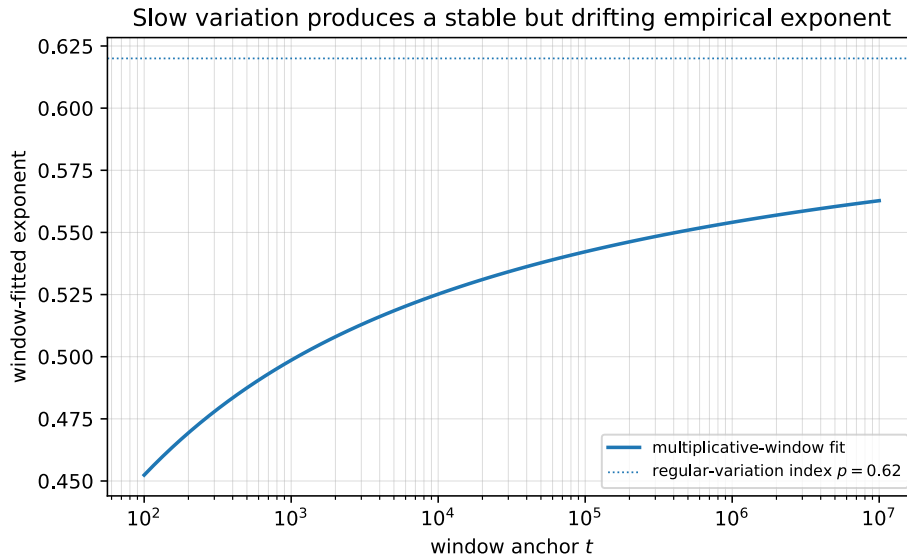


Figure 3: A multiplicative-window regression of $g^{0.62} / \log(e + g)$ converges to its regular-variation index while retaining finite-scale bias.

11 Recursive mechanisms and certified scaling basins

The output characterizations in [Section 9](#) identify the forms of stable scalar profiles. This section derives mechanisms that produce those forms from recursive realization structure. A hierarchy level is not represented by one number alone: its boundary traffic can be decomposed into modes associated with components, interfaces, fanout classes, protocol classes, or other typed contributions. The vector v_k records those modes; the matrix A_k records how one level composes them into the next; and the observation vector u selects the profile reported to the designer.

Recursive positive composition supplies a renormalization mechanism. Projective mixing forgets the initial mode proportions while retaining total amplitude; periodic composition retains a hierarchy phase; vanishing perturbations

create a slowly varying envelope. This yields a derivation of scale behavior from a registered realization mechanism rather than an inference from a fitted line.

Theorem 11.1 (Summable scalar distortion). *Let $b > 1$, $c > 0$, and suppose a positive profile on $g_k = g_0 b^k$ obeys*

$$R(g_{k+1}) = cR(g_k)(1 + \varepsilon_k), \quad \varepsilon_k > -1,$$

with $\sum_{k \geq 0} |\varepsilon_k| < \infty$. Then $\prod_{j < k} (1 + \varepsilon_j)$ converges to a finite positive limit and

$$R(g_k) = \Theta(c^k) = \Theta(g_k^p), \quad p = \log_b c.$$

Under the canonical log-linear interpolation, the profile is regularly varying with index p and in fact has an asymptotically constant reduced amplitude.

Proof. Iteration gives

$$R(g_k) = R(g_0) c^k \prod_{j < k} (1 + \varepsilon_j).$$

Absolute summability, positivity of every factor, and the convergence of $\sum_j \log(1 + \varepsilon_j)$ after a finite prefix imply a positive finite product limit. Since $c^k = (g_k/g_0)^{\log_b c}$, the result follows. $\square$

The vector mechanism retains mode composition and hierarchy phase rather than reducing the recursion to one scalar multiplier.

Definition 11.2 (Positive interface-mode cocycle). *At scales $g_k = g_0 b^k$, let $v_k \in \mathbb{R}_{++}^m$ collect typed interface modes, let $u \in \mathbb{R}_{++}^m$ be the declared observation vector, and write*

$$v_{k+1} = A_k v_k + \eta_k, \quad Y_k = u^\top v_k, \quad z_k = \frac{v_k}{u^\top v_k}.$$

The relative recursion error is

$$\delta_k = \max_i \frac{|(\eta_k)_i|}{(A_k v_k)_i}.$$

Every sufficiently late A_k is nonnegative and maps the cone interior into itself, and each perturbed update remains in the cone interior. The law is asymptotically h -periodic when

$$A_{nh+j} \longrightarrow A^{(j)}, \quad j = 0, \dots, h-1,$$

in any fixed matrix norm. Each limiting phase map preserves the cone interior. With

$$M = A^{(h-1)} \dots A^{(0)},$$

the limiting law is projectively mixing after finite coarse graining when $M^\nu > 0$ entrywise for some fixed integer $\nu \geq 1$.

For $x, y > 0$, Hilbert's projective metric is

$$d_H(x, y) = \log \frac{\max_i x_i / y_i}{\min_i x_i / y_i}.$$

A positive matrix has finite projective diameter and contracts this metric [50, 51].

Theorem 11.3 (Perron–Floquet scaling on the certified mixing class). *Under Theorem 11.2, suppose $v_0 > 0$, $M^\nu > 0$ for some fixed ν , and $\delta_k \rightarrow 0$. Then:*

(i) there is a unique positive u -normalized phase cycle $r_0, \dots, r_{h-1}$ such that

$$z_{nh+j} \rightarrow r_j, \quad r_{j+1} = \frac{A^{(j)} r_j}{u^\top A^{(j)} r_j},$$

for every admitted positive initial trajectory, with phase indices read modulo h ;

(ii) with $c_j = u^\top A^{(j)} r_j$,

$$\frac{Y_{nh+j+1}}{Y_{nh+j}} \rightarrow c_j, \quad \prod_{j=0}^{h-1} c_j = \text{spr}(M);$$

(iii) the profile has

$$Y(g) = Y(g_0)(g/g_0)^p L(g) \psi(\log_b(g/g_0)), \quad p = \frac{\log \text{spr}(M)}{h \log b},$$

where $L \in \text{RV}_0$ and ψ is positive and h -periodic; and

(iv) at hierarchy phase j ,

$$\mathcal{D}_{g_0 b^{nh+j}} Y(x) \longrightarrow x^p \frac{\psi(j + \log_b x)}{\psi(j)}$$

locally uniformly for $x > 0$.

If all $c_j = b^p$, then $Y \in \text{RV}_p$. A nonconstant minimal phase sequence yields asymptotic discrete scale invariance. If

$$\sum_k \left(\|A_k - A^{(k \bmod h)}\| + \delta_k \right) < \infty,$$

then $L(g) \rightarrow L_\infty \in (0, \infty)$.

Proof. Write $[x]_u = x/(u^\top x)$. On the normalized cone section $\{z > 0 : u^\top z = 1\}$, define the unperturbed maps

$$\mathcal{F}_k(z) = \frac{A_k z}{u^\top A_k z}.$$

Because $\delta_k \rightarrow 0$, eventually $\delta_k < 1$. The componentwise relative error bound gives

$$(1 - \delta_k) A_k v_k \leq A_k v_k + \eta_k \leq (1 + \delta_k) A_k v_k.$$

Consequently,

$$d_H(z_{k+1}, \mathcal{F}_k(z_k)) \leq \log \frac{1 + \delta_k}{1 - \delta_k} = O(\delta_k). \quad (11.1)$$

This is the required perturbation bookkeeping. It permits signed η_k provided the relative bound and cone-interior condition hold.

Group the recursion into periods and then into primitive blocks:

$$B_n = A_{nh+h-1} \cdots A_{nh}, \quad C_n = B_{n\nu+\nu-1} \cdots B_{n\nu}.$$

Then $B_n \rightarrow M$ and $C_n \rightarrow M^\nu > 0$ entrywise. After a finite prefix, the νh -step phase-zero maps have uniformly bounded projective diameter and one Birkhoff contraction coefficient $\kappa_{\text{ctr}} < 1$. Let r_0 be the u -normalized Perron direction of M , equivalently of M^ν , and set $e_n = d_H(z_{n\nu h}, r_0)$. Nonnegative cone-interior maps are nonexpansive in Hilbert metric. Applying (11.1) through one finite block, using $C_n \rightarrow M^\nu$ and $[M^\nu r_0]_u = r_0$, gives

$$e_{n+1} \leq \kappa_{\text{ctr}} e_n + \epsilon_n, \quad \epsilon_n \rightarrow 0.$$

Hence $e_n \rightarrow 0$. Propagation through the intervening period and phase maps gives

$$z_{nh+j} \rightarrow r_j, \quad r_{j+1} = [A^{(j)} r_j]_u.$$

Because $[M r_0]_u = r_0$, the cycle closes. Primitivity of M makes r_0 , and therefore the whole phase cycle, unique and independent of v_0 .

The scalar gain satisfies

$$\frac{Y_{k+1}}{Y_k} = u^\top A_k z_k + \frac{u^\top \eta_k}{Y_k}. \quad (11.2)$$

The perturbation term is controlled without assuming cancellation:

$$\left| \frac{u^\top \eta_k}{Y_k} \right| \leq \delta_k u^\top A_k z_k = o(1). \quad (11.3)$$

Matrix and state convergence therefore yield the phasewise gains $c_j > 0$. The cycle relation is $A^{(j)}r_j = c_j r_{j+1}$; multiplying over one period gives

$$Mr_0 = \left(\prod_j c_j \right) r_0.$$

Perron–Frobenius theory implies $\prod_j c_j = \text{spr}(M)$. The profile representation and dilation limits now follow from [Theorems 9.7 and 9.8](#).

For the strengthened conclusion, discard a finite prefix. The normalized trajectory and limiting cycle then lie in a compact subset of the cone interior, where normalization and the positive gain functional are uniformly Lipschitz. If β_n is the sum of matrix and relative errors in the n th νh -block, summability sharpens the recurrence to

$$e_{n+1} \leq \kappa_{\text{ctr}} e_n + C\beta_n, \quad \sum_n \beta_n < \infty.$$

The contracted recurrence gives $\sum_n e_n < \infty$; finite-step propagation then gives

$$\sum_k d_H(z_k, r_{k \bmod h}) < \infty.$$

Equations (11.2)–(11.3) and Lipschitz continuity imply

$$\sum_k \left| \log \frac{Y_{k+1}/Y_k}{c_{k \bmod h}} \right| < \infty.$$

The last clause of [Theorem 9.8](#) makes the slowly varying envelope converge to a positive finite constant. $\square$

Interpretation of the Perron–Floquet exponent. Over one complete hierarchy period, the limiting product M maps a mode mixture back to the same projective phase and multiplies its amplitude by $\text{spr}(M)$. The hierarchy scale grows by b^h during the same period. Equating these two multiplicative changes gives

$$(b^h)^p = \text{spr}(M), \quad p = \frac{\log \text{spr}(M)}{h \log b}.$$

The exponent is therefore the ratio of amplitude growth to scale growth. The phase factors c_j distribute that period gain among individual hierarchy steps, and the slowly varying factor records the residual nonperiodic drift.

Corollary 11.4 (Stationary primitive recursion). *If $A_k \rightarrow A$, A is primitive, and the relative recursion error tends to zero, then*

$$\frac{v_k}{u^\top v_k} \rightarrow r, \quad \frac{Y_{k+1}}{Y_k} \rightarrow \text{spr}(A), \quad Y \in \text{RV}_{\log_b \text{spr}(A)}.$$

If the matrix and relative errors are summable, then $Y(g) \sim Cg^{\log_b \text{spr}(A)}$.

Proof. Apply [Theorem 11.3](#) with $h = 1$. $\square$

Remark 11.5 (Scope of the basin certificate). *The theorem classifies every admitted trajectory in the asymptotically finite-periodic, projectively mixing, finite-mode class with vanishing relative error. It is an in-basin result after membership hypotheses are supplied. It does not say that positivity, finite mode dimension, projective mixing, or periodicity is necessary for an observed regularly varying profile. The scalar equivalence in [Theorem 9.8](#) characterizes output ratios, not latent dynamics. Nonlinear, infinite-mode, stochastic, imprimitive, quasiperiodic, or nonpositive mechanisms remain outside this certificate and may require different theorems.*

12 Arithmetic modes and finite-scale curvature

For a fixed realization and hierarchy, classify boundary events into disjoint internal component classes and an interface class. Additive traffic accounting gives

$$R(g) = \sum_{j=1}^J R_j(g) + R_\partial(g). \tag{12.1}$$

For a netlist hypergraph $H = (V, E)$ and module partition Π_{part} , let

$$\mathcal{C}_H(\Pi_{\text{part}}) = \sum_{M \in \Pi_{\text{part}}} T_H(M)$$

be aggregate terminal incidence. If V is divided into disjoint components V_i , with internal hyperedge sets E_i and interface hyperedges E_∂ , then

$$\mathcal{C}_H(\Pi_{\text{part}}) = \sum_i \mathcal{C}_{H_i}(\Pi_{\text{part}}|_{V_i}) + \mathcal{C}_\partial(\Pi_{\text{part}}).$$

Dividing by $|\Pi_{\text{part}}|$ proves (12.1). Under the sum-of-external-degrees convention, every cut hyperedge contributes once to every module it touches [52, 53]. The profile therefore composes arithmetically; fitted exponents generally do not. In the differential identities below, the interface profile is included as one of the indexed positive modes whenever it is active.

Let $x = \log g$ and let each positive mode $R_j(e^x)$ be twice differentiable. Define its local slope and traffic share by

$$p_j^{\text{loc}}(g) = \frac{d \log R_j(g)}{d \log g}, \quad \omega_j(g) = \frac{R_j(g)}{\sum_k R_k(g)}.$$

The additive identity permits an exact differential decomposition. Writing $x = \log g$, each mode obeys $R'_j(x) = p_j^{\text{loc}}(x) R_j(x)$ and the total profile is $R = \sum_j R_j$. The traffic share $\omega_j = R_j/R$ therefore acts as a probability weight over active modes. Differentiating once gives the weighted mean of local exponents; differentiating again separates curvature created by differences among modes from curvature already present within each mode.

Theorem 12.1 (Mode-mixture geometry). *For $R(g) = \sum_j R_j(g)$,*

$$p_{\text{loc}}(g) = \sum_j \omega_j(g) p_j^{\text{loc}}(g),$$

and

$$\frac{dp_{\text{loc}}}{d \log g} = \text{Var}_{\omega(g)}(p_j^{\text{loc}}(g)) + \sum_j \omega_j(g) \frac{dp_j^{\text{loc}}(g)}{d \log g}.$$

Proof. Differentiate $\log R$ to obtain the first identity. With $x = \log g$,

$$\omega'_j = \omega_j(p_j^{\text{loc}} - p_{\text{loc}}).$$

Differentiating $p_{\text{loc}} = \sum_j \omega_j p_j^{\text{loc}}$ and collecting terms gives the variance and drift decomposition. $\square$

For pure modes $R_j = a_j g^{p_j}$, the within-mode drift vanishes:

$$p_{\text{loc}} = \sum_j \omega_j p_j, \quad \frac{dp_{\text{loc}}}{d \log g} = \text{Var}_{\omega}(p_j) \geq 0.$$

Finite-scale curvature is thus a structural observable of heterogeneous crossover, not merely regression noise.

Corollary 12.2 (No exact heterogeneous scalar). *A positive finite sum $\sum_j a_j g^{p_j}$ equals one power law on an open interval if and only if all active exponents p_j are equal.*

Proof. One power has zero log-curvature. By Theorem 12.1, pure modes have zero curvature exactly when their weighted exponent variance is zero. $\square$

Theorem 12.3 (Dominant regularly varying mode). *Let $R_j \in \text{RV}_{p_j}$ be a finite collection of positive functions and let $p_\star = \max_j p_j$. Then*

$$\sum_j R_j \in \text{RV}_{p_\star}.$$

Every mode with exponent below $p_\star$ becomes asymptotically negligible relative to a positive mode with exponent $p_\star$.

Proof. Write $R_j(g) = g^{p_j} \ell_j(g)$. Ratios of regularly varying functions with distinct indices satisfy $R_j/R_k \rightarrow 0$ when $p_j < p_k$ [54]. A finite sum of positive slowly varying functions is slowly varying; factor out $g^{p_\star}$. $\square$

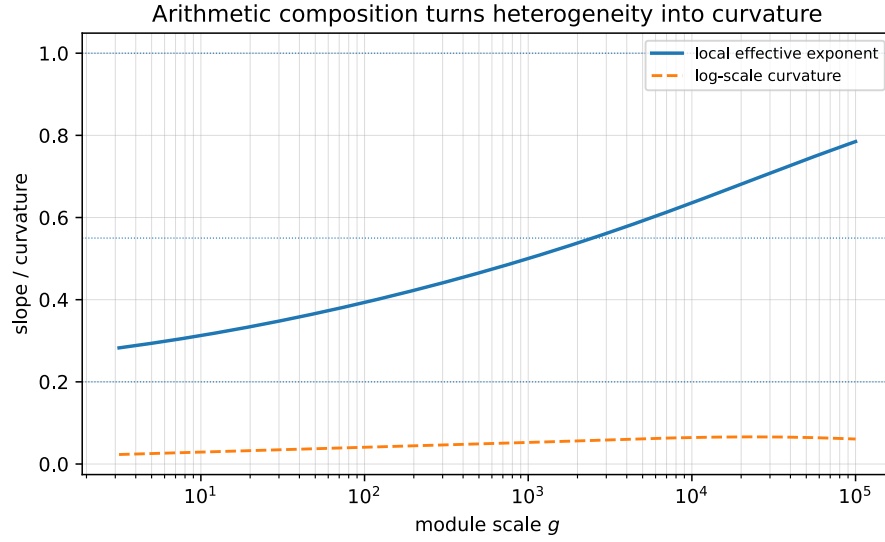


Figure 4: For pure modes, the whole-profile local slope is a traffic-weighted convex mixture and its log-curvature is the weighted exponent variance.

13 Interface reach and terminal-boundary crossover

Let E_∂ be the inter-component hyperedge set. At a size- g hierarchy level, let $\kappa_g(e)$ be the number of modules touched by hyperedge e . Under sum-of-external-degrees accounting, the exact interface profile is

$$R_\partial(g) = \frac{g}{G} \sum_{e \in E_\partial} \kappa_g(e) \mathbf{1}[\kappa_g(e) \geq 2]. \quad (13.1)$$

Interface-net count alone is not the multiscale variable; pin occupancy and hierarchy reach are part of the accounting.

For a two-pin net, $\kappa_g(e) = 2$ exactly while its endpoints remain separated. If $N_\partial^\geq(g)$ counts such nets, then

$$R_\partial(g) = \frac{2g}{G} N_\partial^\geq(g).$$

If $N_\partial^\geq(g) \asymp G^\alpha g^{-\vartheta} \ell(g)$, the interface mode has index $p_\partial = 1 - \vartheta$ up to the slowly varying term. Against a core $R_{\text{core}}(g) = \Theta(g^p)$ at $g = \Theta(G^\beta)$, it is relevant when

$$\alpha > 1 - \beta(p_\partial - p).$$

Uniform random two-pin endpoints give $p_\partial = 1$ away from terminal saturation.

For an r -pin hyperedge whose pins independently choose among $m = G/g$ equal modules,

$$\mathbb{E}[\kappa_g(e)] = m \left[1 - \left(1 - \frac{1}{m} \right)^r \right],$$

and all pins occupy one module with probability m^{1-r} . Hence

$$\mathbb{E}[\kappa_g(e) \mathbf{1}(\kappa_g(e) \geq 2)] = m \left[1 - \left(1 - \frac{1}{m} \right)^r \right] - m^{1-r}.$$

This is a fanout-resolved null model, not a placement model. A global broadcast hyperedge instead touches every module and contributes one unit to the level mean. If $K = \Theta(G^\alpha)$ global nets are present, then

$$R_{\text{broadcast}}(g) = \Theta(G^\alpha),$$

which dominates a core at $g = \Theta(G^\beta)$ when

$$\alpha > \beta p.$$

Clocks, resets, and coherence fabrics must therefore retain their actual fanout profiles rather than use a two-pin surrogate.

Definition 13.1 (Declared cap response). *A cap response is a continuous nondecreasing function $\Phi : (0, \infty) \rightarrow (0, 1]$ satisfying*

$$\Phi(z)/z \rightarrow 1 \quad (z \downarrow 0), \quad \Phi(z) \rightarrow 1 \quad (z \rightarrow \infty).$$

For an interior profile I and declared terminal budget B , define

$$\mathcal{C}_B[I](g) = B\Phi(I(g)/B).$$

The hard cap is $\Phi(z) = \min\{z, 1\}$; a soft rational cap is $\Phi(z) = z/(1+z)$.

Theorem 13.2 (Boundary crossover). *Let $p > 0$, let $B_n \rightarrow \infty$, and choose $g_n \rightarrow \infty$ with $I(g_n)/B_n \rightarrow 1$.*

(i) *If $I \in \text{RV}_p$, then*

$$\frac{\mathcal{C}_{B_n}[I](g_n x)}{B_n} \longrightarrow \Phi(x^p)$$

locally uniformly for $x > 0$. The hard-cap limit is $\min\{x^p, 1\}$.

(ii) *If instead*

$$I(g) = g^p L(g) \psi(\log_b g),$$

where $L \in \text{RV}_0$ and ψ is positive, continuous, and h -periodic, then every subsequence satisfying $\log_b g_n \bmod h \rightarrow s$ has the phase-indexed limit

$$\frac{\mathcal{C}_{B_n}[I](g_n x)}{B_n} \longrightarrow \Phi\left(x^p \frac{\psi(s + \log_b x)}{\psi(s)}\right).$$

Proof. Uniform convergence for regular variation gives $I(g_n x)/I(g_n) \rightarrow x^p$ locally uniformly. Combine it with $I(g_n)/B_n \rightarrow 1$ and continuity of Φ . In the finite-phase branch, slow variation cancels while the periodic factor retains the limiting hierarchy phase. $\square$

Corollary 13.3 (Crossover slope). *In the regularly varying branch of Theorem 13.2, additionally suppose*

$$\frac{gI'(g)}{I(g)} \rightarrow p$$

and $\Phi \in C^1$ near x^p . Then the local logarithmic slope at $g_n x$ converges to

$$pE_\Phi(x^p), \quad E_\Phi(z) = \frac{z\Phi'(z)}{\Phi(z)}.$$

For the hard cap, the statement holds away from $x = 1$, with one-sided slopes at the kink.

Theorem 13.4 (Partial converse from phase-rich cap collapse). *Fix $p > 0$ and let $B_n > 0$. Let I be positive and eventually nondecreasing, let $g_n \uparrow \infty$ satisfy $g_{n+1}/g_n \rightarrow 1$, and suppose $I(g_n)/B_n \rightarrow 1$. Let J be an open multiplier interval containing 1. Assume Φ restricts to a homeomorphism on an interval containing $\{x^p : x \in J\}$ and all sufficiently late normalized arguments. If*

$$\frac{\mathcal{C}_{B_n}[I](g_n x)}{B_n} \longrightarrow \Phi(x^p)$$

locally uniformly for $x \in J$, then $I \in \text{RV}_p$.

Proof. Continuity of Φ^{-1} gives

$$\frac{I(g_n x)}{B_n} \rightarrow x^p, \quad \frac{I(g_n x)}{I(g_n)} \rightarrow x^p$$

locally uniformly on J . For arbitrary $t \rightarrow \infty$, bracket t between adjacent g_n, g_{n+1} . Eventual monotonicity, the dense-anchor condition, and slightly smaller and larger multipliers in J squeeze $I(tx)/I(t)$ to x^p for $x \in J$. Finite products and reciprocals of an open interval containing 1 cover $(0, \infty)$, so the dilation limit holds for every positive multiplier. $\square$

The cap shape is a declared model component represented by Φ . A purely dyadic grid samples a period-one function of $\log_2 g$ at one phase and cannot meet the dense-anchor premise of the converse. The two-sided converse also excludes the hard cap because it is noninjective after saturation; a one-sided version can use an unsaturated interval. Terminal budgets and pinned boundary conditions are coordinates of the resource contract ρ , not measurement noise.

14 Finite-scale observability and failure taxonomy

At one level, let per-module traffic be a positive random variable T . Jensen's inequality gives

$$\log \mathbb{E}[T] - \mathbb{E}[\log T] \geq 0.$$

Thus the logarithm of the level mean and the mean of per-module logarithms are different estimators even before fitting across scales. For $\text{CV}(T) \ll 1$ and the usual moment regularity,

$$\log \mathbb{E}[T] - \mathbb{E}[\log T] = \frac{1}{2} \text{CV}(T)^2 + o(\text{CV}(T)^2).$$

Heavy-tailed module traffic can make the discrepancy large.

Repeated composition can also contract variation around an expected profile. Fix levels $g_1, \dots, g_J$ and write

$$Y_j = \sum_{r=1}^N Z_{rj} > 0,$$

where the contributor vectors Z_r are independent, although coordinates within one vector may be arbitrarily dependent across levels. Suppose Z_{rj} lies in an interval of width Δ_{rj} . For $x_j = \log g_j$, let a_j be the OLS slope weights, so that

$$\hat{p} = \sum_{j=1}^J a_j \log Y_j.$$

Theorem 14.1 (Concentration of a fitted slope). *Let*

$$\mu_j = \mathbb{E}[Y_j] \geq \mu_{\min} > 0, \quad V = \max_j \sum_{r=1}^N \Delta_{rj}^2,$$

and define

$$p_\star = \sum_{j=1}^J a_j \log \mu_j.$$

For $0 < \delta < 1$, set

$$t = \sqrt{\frac{V}{2} \log \frac{2J}{\delta}}.$$

If $t \leq \mu_{\min}/2$, then with probability at least $1 - \delta$,

$$|\hat{p} - p_\star| \leq \frac{2\|a\|_1}{\mu_{\min}} \sqrt{\frac{V}{2} \log \frac{2J}{\delta}}.$$

Proof. Hoeffding's inequality and a union bound control every $|Y_j - \mu_j|$ by t . On that event $Y_j \geq \mu_{\min}/2$, and the mean-value theorem gives

$$|\log Y_j - \log \mu_j| \leq \frac{2}{\mu_{\min}} |Y_j - \mu_j|.$$

Apply the triangle inequality to the linear slope functional. $\square$

For comparable contributors averaged with weights $1/m$, the width proxy is $V = O(1/m)$. For unequal normalized contribution weights α_i , the corresponding effective count is

$$m_{\text{eff}} = \frac{1}{\sum_i \alpha_i^2}.$$

Repeated IP, common clocks, and shared coherence protocols can make m_{eff} far smaller than the raw block count.

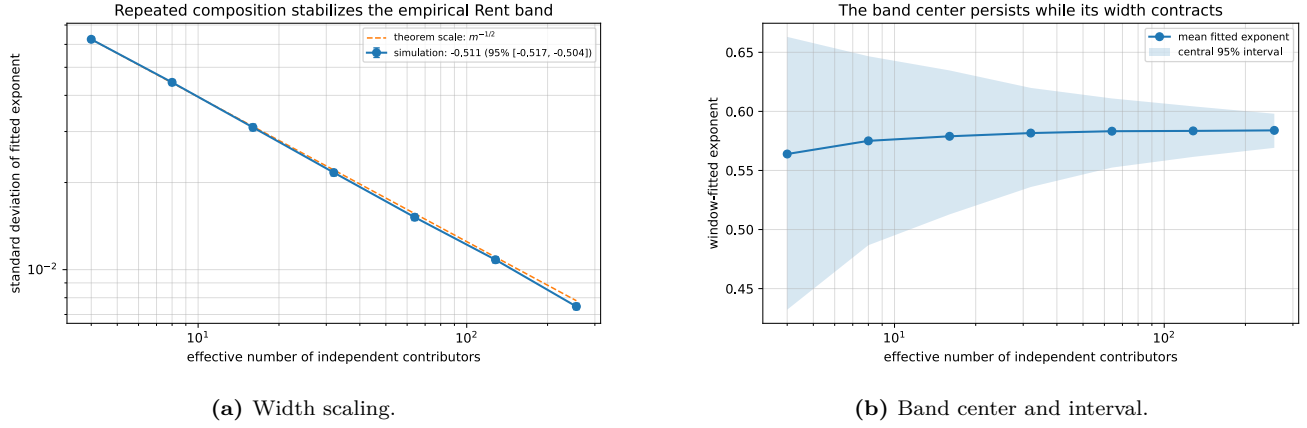


Figure 5: Under the bounded-independent composition model, a persistent expected center coexists with a shrinking distribution of fitted slopes.

Dynamical hypothesis	Profile consequence	Observable diagnostic	Failure meaning
Autonomous mixing cocycle	$g^p L(g)$	Mode direction and adjacent slope converge	Rejects this certificate, not all scaling mechanisms
Summable autonomous distortion	$Cg^p(1 + o(1))$	Reduced amplitude converges	Rejects summable correction
h -periodic mixing cocycle	$g^p L(g)\psi(\log_b g)$	Phase slopes converge to an h -cycle	Rejects the finite-period certificate
Regular interior plus declared cap	$B\Phi((g/g_*)^p)$	Normalized crossover collapse	Rejects the registered cap model
Finite-phase interior plus cap	Phase-indexed cap curves	Collapse depends on log phase	One-phase samples cannot certify membership
No convergent or periodic gain law	No class asserted	Adjacent ratios neither converge nor cycle	Outside this sufficient taxonomy

Table 4: Mechanism, consequence, and falsifier are separated. Observing a consequence does not prove latent basin membership.

Theorem 14.2 (Stable empirical generalized law). *Let the expected profile be $\mu(g) = g^p \ell(g)$ with ℓ slowly varying, and measure on a fixed multiplicative window $g_j(t) = t\lambda_j$. Suppose the independent bounded-composition conditions of Theorem 14.1 hold and its stochastic bound is $O(m_{\text{eff}}^{-1/2})$. Then*

$$\hat{p}(t, m_{\text{eff}}) = p + b_\ell(t) + O_{\mathbb{P}}(m_{\text{eff}}^{-1/2}),$$

where

$$b_\ell(t) = \sum_j a_j \log \frac{\ell(t\lambda_j)}{\ell(t)} \rightarrow 0.$$

Consequently, $\hat{p} \rightarrow p$ in probability when the window moves outward and $m_{\text{eff}} \rightarrow \infty$.

Proof. The expected-profile slope is $p + b_\ell(t)$ by Theorem 10.1; the stochastic deviation follows from Theorem 14.1. $\square$

This is a conditional mechanism for an empirical band: slow variation controls its finite-window bias, mode crossover controls curvature, and effective independent composition controls width. It does not identify the mechanism behind a historical fitted band or make its center universal.

Trees, lattices, expanders, butterflies, and heterogeneous systems calibrate different rows of this taxonomy: bounded tree profiles can have index 0; d -dimensional lattices have surface-growth index $(d-1)/d$; expanders have linear boundary growth; butterfly networks carry an index-one $1/\log g$ correction; and heterogeneous systems combine several modes with an explicit interface profile. These examples calibrate distinct scaling classes; assigning a computation to a class requires the corresponding structural hypotheses. At all realizable sizes, the measured profile, hierarchy convention, correction term, and estimator remain primary; an asymptotic index can be operationally remote.

14.1 Deterministic mechanism checks

The phase-rich deterministic check uses the declared scalar observation vector $u = (1, 1)^\top$ and

$$A^{(0)} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \quad A^{(1)} = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix}.$$

For $M = A^{(1)}A^{(0)}$, $\text{spr}(M) = 4 + \sqrt{15}$ and

$$p = \frac{\log(4 + \sqrt{15})}{2 \log 2} \approx 1.488455.$$

The reference script `artifacts/renormalization_validation_v1.py` evaluates every level $k = 0, \dots, 64$ for four positive initial mixtures. The recorded phase gains are $c_0 = 2.4364916731$ and $c_1 = 3.2312785769$, while $2^p = 2.8058837015$ differs from both. Across late samples, the maximum Hilbert distance to the phase cycle is 4.44×10^{-16} , the maximum adjacent-ratio error is 8.88×10^{-16} , and the relative tail span of the phase-reduced amplitude is 8.61×10^{-15} . The product $c_0 c_1$ agrees with $4 + \sqrt{15}$ to recorded precision.

The same program registers quasiperiodic, nonnegligibly forced, nonmixing, and bounded-no-exponent controls under `artifacts/renormalization_results_v1/`. These computations verify the bookkeeping and observables. The proofs establish the theorem statements, while physical-netlist basin membership is determined by the registered structural hypotheses and diagnostics.

15 Public-netlist diagnostics and mode-resolved attribution

The scaling theory distinguishes three questions: the output law of a profile, a mechanism that can generate that law, and a measurement instrument that observes a finite realization. This section addresses the third. Two deterministic hierarchy instruments instantiate reproducible, hierarchy-conditioned scale profiles and test finite-scale identities, estimator dependence, and cross-hierarchy transfer.

The diagnostics use ten named BLIF snapshots from the public EPFL combinational benchmark suite [49], ranging from 374 to 57,503 vertices. Every reported value belongs to a complete measurement tuple

(netlist, parser, hierarchy, traffic convention, scale window, estimator).

This tuple is the empirical object. It permits reproducible comparison of hierarchy and estimator effects while keeping the logical netlist fixed.

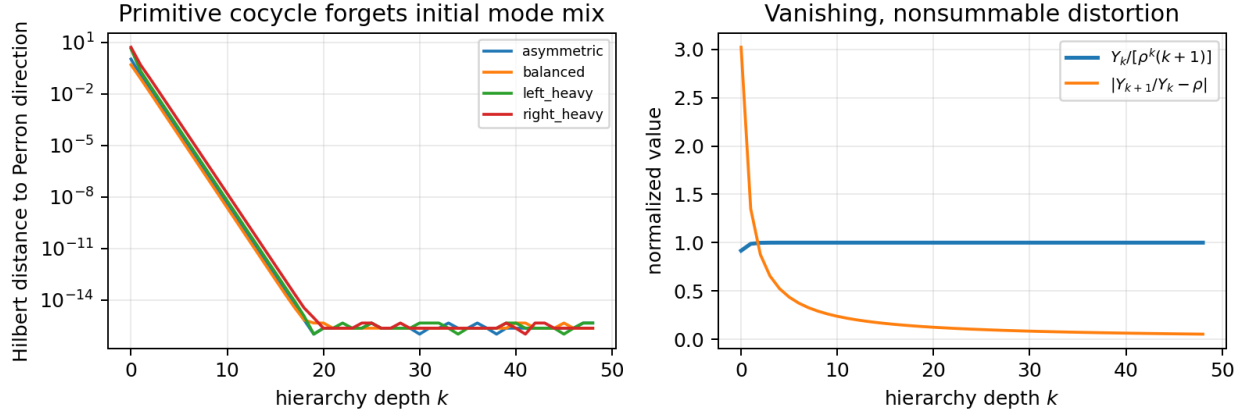
Two deterministic hierarchy instruments are used. The first computes one reverse-Cuthill–McKee (RCM) ordering and divides it into contiguous, size-regular, power-of-two module counts. It is run on all ten files. This instrument is geography-like but is not cut optimized. The second is a recursive RCM hierarchy: each module above a declared threshold is reordered locally before bisection, while inherited order is preserved below the threshold. It is run on the barrel shifter and multiplier. Bidirectional cross-hierarchy transfer is evaluated only on the multiplier. The paired runs vary hierarchy construction while holding parser and traffic accounting fixed.

15.1 Registered measurement and reproduction protocol

The measurement contract is registered at the following resolution.

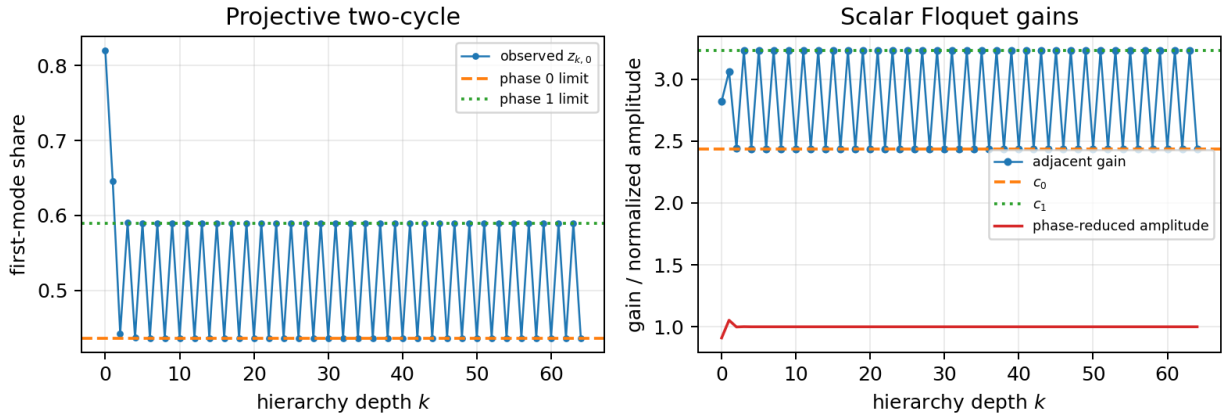
- (a) **Dataset identity.** The exact inputs are `adder.blif`, `arbiter.blif`, `bar.blif`, `div.blif`, `i2c.blif`, `log2.blif`, `multiplier.blif`, `router.blif`, `sqrt.blif`, and `square.blif` under `artifacts/source_netlists/`. File hashes are recorded in the root SHA-256 manifest. Circuit names denote these snapshots, not every version of the corresponding design.
- (b) **Representation.** The BLIF parser creates explicit primary-input, gate-output, and primary-output vertices and one hyperedge for every signal incident on at least two distinct vertices. The implementation constructs a weighted sparse graph projection, sums duplicate CSR entries, and sorts each CSR row by column index before RCM. The same canonicalization is applied to every recursively induced subgraph. This fixes neighbor-visitation order, including degree ties, in the supported software environment.
- (c) **Hierarchy.** Fixed RCM forms one ordering and splits it into 1, 2, 4, $\dots$ nearly equal contiguous blocks until every block has at most eight vertices. Recursive RCM bisects each current module, recomputing its local RCM order above 1,024 vertices and preserving inherited order below that threshold, again stopping at eight vertices.

Autonomous Perron basin and slowly varying envelope



(a) Autonomous projective forgetting.

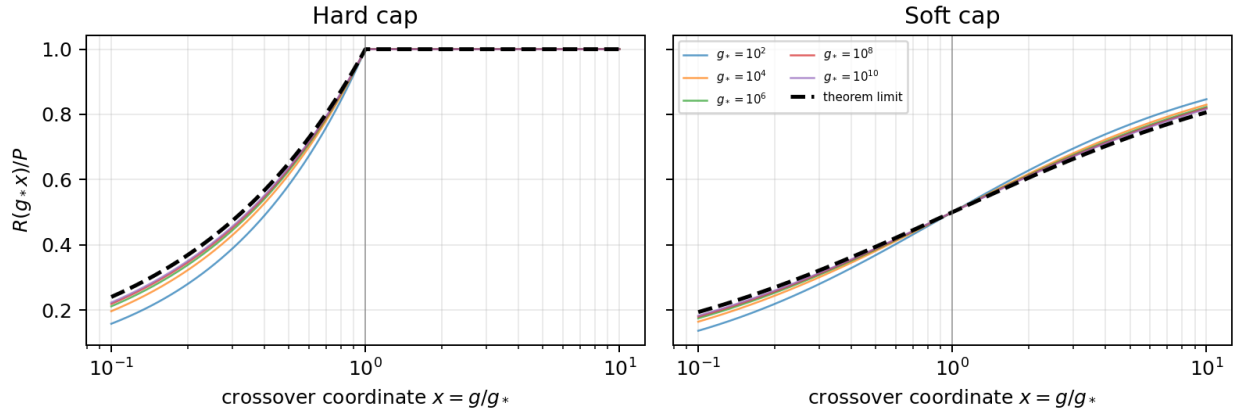
Periodic positive cocycle: one exponent, persistent hierarchy phase



(b) Periodic Perron–Floquet cycle.

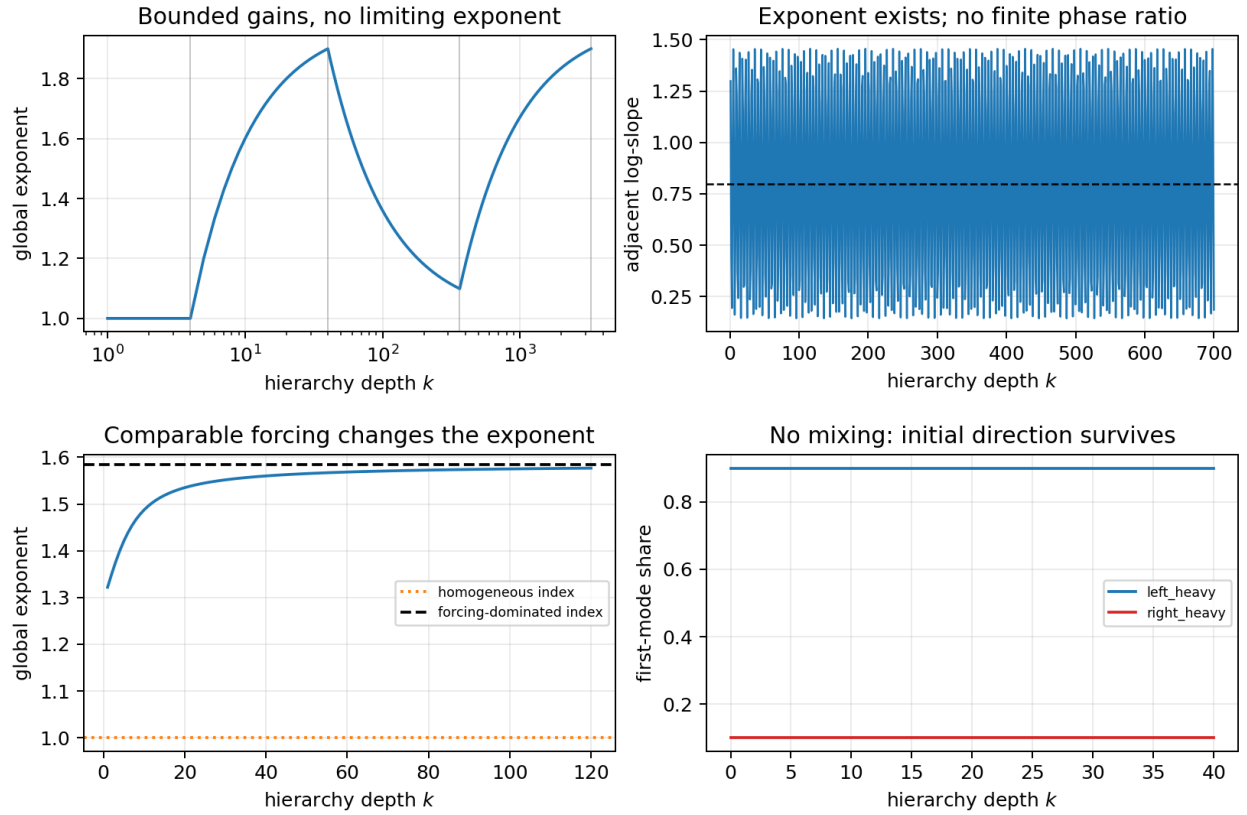
Figure 6: Deterministic checks of autonomous projective forgetting and finite-period phase retention.

Boundary crossover collapse retains the registered cap response



(a) Registered cap responses.

Negative controls delimit the universality basin



(b) Required negative controls.

Figure 7: Deterministic checks of boundary-cap collapse and registered failure modes.

Circuit	Vertices	Fixed-hierarchy $\hat{p}$	Scatter $\hat{p}$
router	374	0.095	0.294
adder	1,405	0.000	-0.029
i2c	1,646	0.807	0.820
barrel shifter	3,599	0.756	0.805
arbiter	12,224	0.976	0.978
square	18,678	0.843	0.818
square root	24,810	0.941	0.936
multiplier	27,318	0.909	0.910
logarithm	32,124	0.864	0.850
divider	57,503	0.947	0.949

Table 5: Diagnostics under the deterministic fixed hierarchy. Values are conditioned on the registered estimator and hierarchy; they are not intrinsic Rent exponents.

- (d) **Traffic and scale window.** A non-I/O hyperedge contributes once to every module it touches whenever it crosses a module boundary. The fanout classes are 2, 3–8, 9–32, and above 32 pins. Unweighted ordinary least squares fits $\log R$ against $\log g$ only on levels with positive internal traffic and

$$16 \leq g \leq G/4.$$

The level-mean estimator uses $\log(\text{mean } T)$; the scatter estimator uses $\text{mean}(\log \max\{T, 1\})$.

- (e) **Mode attribution.** A mode law is fitted only when that mode is positive on at least three retained levels. Profile relative RMSE is

$$\left[\frac{1}{m} \sum_i \left(\frac{\hat{R}_i - R_i}{R_i} \right)^2 \right]^{1/2}.$$

Slope and curvature RMSEs are ordinary root-mean-square differences after second-order finite differences on the log-scale grid.

- (f) **Cross-hierarchy transfer.** The four source-hierarchy mode laws are frozen. One total-amplitude anchor is permitted at the largest common target scale; every other target level is a prediction. Thus transfer cannot refit mode exponents or introduce level-specific corrections.
- (g) **Determinism and records.** The RCM paths invoke no random generator, so there is no empirical seed parameter. With sparse-ordering canonicalization, two independent reruns produced byte-identical CSV, JSON, and PNG outputs. Generated PDFs differ only in creation metadata and are compared by rendered content rather than raw bytes. Environment versions are pinned in `artifacts/requirements.txt`. Executable methods are in `artifacts/epfl_spectrum_analysis.py`, `artifacts/epfl_crg_validation.py`, and `artifacts/cross_hierarchy_mode_test.py`; level tables, summaries, and transfer records are under `artifacts/epfl_results_v2/` and `artifacts/results_epfl_v3/`.

15.2 Hierarchy- and estimator-conditioned scalar fits

The ten fitted values span nearly the whole interval from zero to one under a single registered procedure. This dispersion is descriptive of the retained snapshots and scale windows. It is not a population estimate for circuits in general. The level-mean and scatter estimators differ most strongly on the router, where their fitted slopes are 0.095 and 0.294, respectively, consistent with the finite-level Jensen gap in [Section 14](#).

Hierarchy choice is also material. For the multiplier, the fixed ordering gives $\hat{p} \approx 0.909$, whereas recursive local RCM gives $\hat{p} \approx 0.831$.

Versioned diagnostic note. An earlier measurement contract reported the multiplier pair $0.901 \rightarrow 0.839$. The values $0.909 \rightarrow 0.831$ used here are from the controlling registered protocol, which additionally fixes sparse-row canonicalization, hierarchy thresholds and stopping rules, and the fit window $16 \leq g \leq G/4$. The retained records do not isolate the numerical change to one pipeline modification. The two pairs therefore remain outputs of different versioned measurement contracts; neither is interpreted as an intrinsic hierarchy-independent exponent.

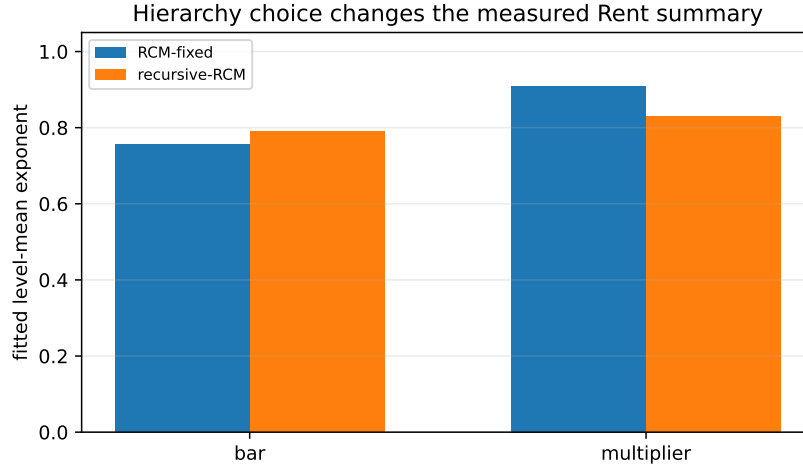


Figure 8: The multiplier scalar fit changes under two deterministic hierarchy instruments even though parser and traffic accounting remain fixed.

15.3 Exact mode identity, finite-scale failure, and transfer

At each retained level, internal incidence is decomposed into the four fanout classes. The arithmetic identity

$$R(g) = \sum_{j=1}^4 R_j(g)$$

holds with zero numerical residual under both multiplier hierarchies. Two scientifically different tests follow.

First, fit one pure law $a_j g^{p_j}$ to each measured fanout mode. Under the fixed hierarchy the fitted mode exponents are approximately

$$(0.931, 0.902, 0.508, 0.231),$$

whereas recursive local RCM gives

$$(0.891, 0.754, 0.696, 0.192).$$

The sum of these global mode fits does not closely reproduce the whole finite-scale curve. Profile relative RMSE is 0.135 under fixed RCM and 0.0478 under recursive RCM. The variance-only pure-mode approximation has slope RMSEs 0.265 and 0.0555, respectively. Hence the real fanout modes are themselves scale dependent; the pure-power specialization of [Theorem 12.1](#) is falsified as a complete finite-scale model.

Second, retain the measured local slopes

$$p_j^{\text{loc}}(g) = \frac{d \log R_j(g)}{d \log g}$$

and their drift. With

$$\omega_j(g) = \frac{R_j(g)}{\sum_k R_k(g)},$$

the exact identities are

$$p_{\text{loc}}(g) = \sum_j \omega_j(g) p_j^{\text{loc}}(g),$$

$$\frac{dp_{\text{loc}}}{d \log g} = \text{Var}_{\omega(g)}(p_j^{\text{loc}}) + \sum_j \omega_j(g) \frac{dp_j^{\text{loc}}}{d \log g}.$$

Using modes measured under the same hierarchy reconstructs the multiplier slope with RMSE 0.00935 under fixed RCM and 0.00318 under recursive RCM; curvature RMSEs are 0.0497 and 0.00876, respectively. This validates the traffic decomposition and numerical implementation of the algebraic identity. Because $R = \sum_j R_j$, the reconstruction is exact; its scientific role is to separate exponent dispersion from within-mode drift.

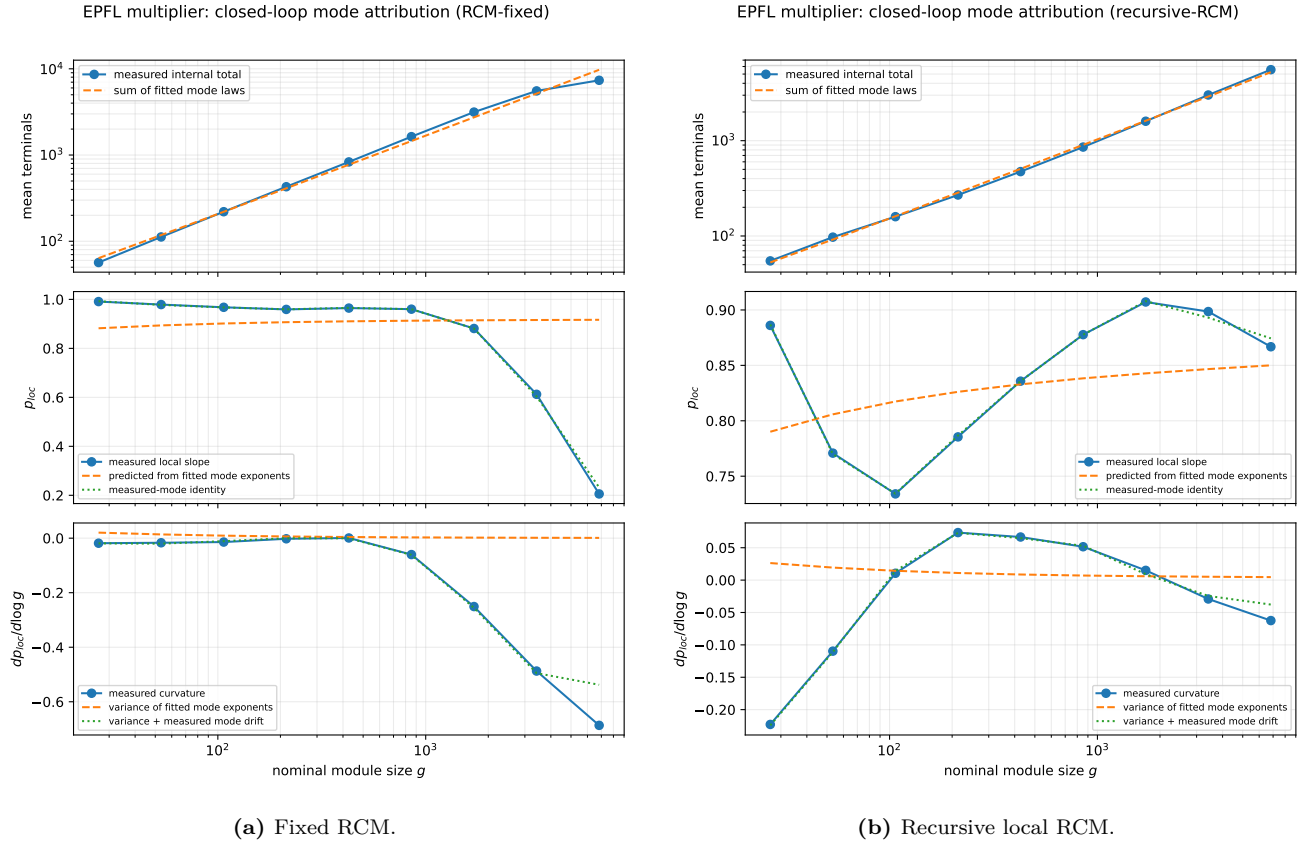


Figure 9: The global pure-mode approximation is incomplete. The measured-mode identity, including within-mode drift, tracks slope and curvature under each hierarchy.

The cross-hierarchy transfer test is independently falsifiable. Fixed-RCM mode laws transferred to recursive RCM have profile relative RMSE 0.175 and local-slope RMSE 0.0864. In the reverse direction the errors are 0.194 and 0.261. Fanout classes remain identifiable, but their effective scale laws are hierarchy conditioned. A hierarchy is therefore an active part of the measurement rather than a passive coordinate change.

Internal and pad-inclusive profiles also separate at the largest scales, consistent with a terminal-boundary crossover. This observation does not identify a unique cap response Φ ; it only shows that terminal convention must be reported with the hierarchy, fanout accounting, and estimator.

15.4 Interpretation of the diagnostic

The registered experiment establishes four empirical facts. First, the measurement is deterministic and reproducible in the pinned environment. Second, level-mean and scatter estimators can assign materially different slopes to the same hierarchy. Third, changing the hierarchy construction can change both the scalar fit and the mode laws while leaving parser and traffic accounting unchanged. Fourth, the exact mode-mixture identity closes numerically when its modes and local slopes are measured under the same hierarchy.

These facts locate the diagnostic inside CRG. The BLIF snapshots are logical realizations, so their measured objects are topology- and hierarchy-level profiles. Fixed and recursive RCM are two admissible topology instruments; their separation is a direct measure of hierarchy sensitivity. The profiles belong to one realization and hierarchy at a time, and fixed-hierarchy additivity applies to mode incidence inside that realization.

The pure-mode fit and measured-mode reconstruction answer different questions. Failure of the pure-mode transfer demonstrates that one global power per fanout class does not describe the complete finite-scale profile. Closure of the measured-mode identity verifies the arithmetic decomposition and separates between-mode exponent dispersion from within-mode drift. The dyadic scale grid samples one logarithmic phase; shifted and nondyadic hierarchies supply the additional phase information needed to resolve a periodic factor. The largest scales are terminal affected, so finite-scale mode shares, drift, boundary response, and transfer error are the primary empirical

outputs.

The physical-design stage enters later in the CRG chain. Placement, routing, buffering, timing, activity, power, area, and calibrated fabric service map these logical realization profiles into technology-conditioned witnesses. The present diagnostic supplies the hierarchy-conditioned logical input to that stage.

Part III

Technology-conditioned realization and certified design

16 From attainable geometry to physical decisions

The preceding parts construct the legal realization family and its decision-sufficient geometries before physical technology is applied. This part completes the computation-to-technology chain. A technology supplies capabilities, admissible embeddings, service laws, and costs; the design problem is to determine which legal realization and embedding minimize the declared physical objective while preserving every alternative required by the future decision class.

CRG and existing design tools occupy complementary mathematical roles. Timeloop/Accelergergy, MAESTRO, ZigZag, CoSA, Gemini, and physical implementation flows generate, evaluate, and refine constructive witnesses [37, 43, 38, 39, 40, 41]. CRG supplies the computation-conditioned realization bundle, typed joint obligations, scope-matched converses, and exact retention conditions under which those witness sets may be reduced. The combined flow is closed-loop: search and physical tools return realizations to the geometry, and the geometry determines which further searches and physical measurements can change the decision.

Fix a terminalized computation F , a resource contract ρ , and an admissible hierarchy $\mathcal{T}$. Let $\mathfrak{R}_F^\rho(\mathcal{T})$ be the legal-realization space and $\sigma_{\mathcal{T}}(P) = z = (u, \mu)$ its typed signature, where u records local obligations and μ records interface state, traffic, causal, and activity coordinates. Different realizations can share one signature, so a detailed physical objective need not factor through z .

Definition 16.1 (Technology-conditioned physical objective). *For technology τ , let $\text{Emb}_\tau(P, \mathcal{T})$ be the legal embeddings of realization P . Define*

$$\text{OPT}_\tau^\rho(F) = \inf_{\substack{\mathcal{T} \in \text{Dec}_\rho(F), P \in \mathfrak{R}_F^\rho(\mathcal{T}), \\ \phi \in \text{Emb}_\tau(P, \mathcal{T})}} \Phi_\tau(P, \phi).$$

When the objective observes only the typed signature and embedding, write $\Phi_\tau(P, \phi) = \hat{\Phi}_{\tau, \mathcal{T}}(\sigma_{\mathcal{T}}(P), \phi)$. Timing, routing, clocking, switching, thermal, and reliability objectives may distinguish realizations in the same signature fiber.

Definition 16.2 (Price update and capability update). *Fix a computation F , a resource contract ρ , and a technology $\tau = (\text{Emb}_\tau, \Phi_\tau)$. A price update replaces Φ_τ by another objective $\Phi_{\tau'}$ while preserving the legal realization family and the embedding relation:*

$$\mathcal{P}_F^\rho(\mathcal{T}) \text{ unchanged,} \quad \text{Emb}_{\tau'}(P, \mathcal{T}) = \text{Emb}_\tau(P, \mathcal{T}).$$

A capability update changes an allowed primitive, hierarchy, storage mode, schedule, carrier, or embedding relation and therefore may replace ρ by ρ' or Emb_τ by a different admissibility relation.

Proposition 16.3 (Repricing and branch reopening). *Under a price update, the pre-technology CRG objects $\mathcal{P}, \mathcal{R}, \Gamma$, and $\mathcal{K}$ are unchanged; the updated decision is obtained by re-evaluating the new objective on the retained object sufficient for its observation class. Under a capability update, the legal realization or embedding family may change. A retention certificate transfers to the updated problem exactly when the updated legal family contributes no additional decision-relevant point and the new objective remains inside the observation class preserved by the retained family.*

Proof. A price update changes neither legality nor attainable signatures, so the abstraction ladder is identical and only the response functional changes. A capability update can add or remove realizations or embeddings, hence can change the exact image and every downstream quotient. If the updated family adds no point that improves a decision in the preserved class, the relevant quotient remains equal and the Safe-Commitment Law transfers. Conversely, if the updated quotient differs, its reconstruction theorem supplies a future monotone objective or nonnegative price that distinguishes the old and new families; the affected branch must be reopened. $\square$

Across hierarchies, signature dimensions and legality rules differ. The global design space is therefore the stratified disjoint union

$$\mathfrak{D}_F^\rho = \bigsqcup_{\mathcal{T} \in \text{Dec}_\rho(F)} \{\mathcal{T}\} \times \mathfrak{R}_F^\rho(\mathcal{T}),$$

not one global convex body. Convex support arguments apply within a fixed hierarchy and observation class; cross-hierarchy selection retains the stratum label and its physical embedding constraints.

16.1 Pareto sufficiency and price-conditioned regimes

Proposition 16.4 (Signature dominance). *Fix a hierarchy and an objective that factors through the signature and is coordinatewise nondecreasing. If $z_1 \leq z_2$, then z_2 cannot have lower cost than z_1 . Hence only the Pareto-minimal signature boundary can be preferred by that observation class.*

Proof. Monotonicity gives $\widehat{\Phi}_{\tau, \mathcal{T}}(z_1, \phi) \leq \widehat{\Phi}_{\tau, \mathcal{T}}(z_2, \phi)$ whenever both signatures are evaluated under the same admissible embedding conditions. The claim does not compare realization fibers when the objective observes more than z . $\square$

For a fixed hierarchy and a nonnegative additive price vector w , a point z of the convex upper body $\mathcal{K}_F^\rho(\mathcal{T})$ is optimal exactly when w lies in its lower normal cone

$$N_{\mathcal{K}}^-(z) = \{w \geq 0 : \langle w, y - z \rangle \geq 0 \text{ for all } y \in \mathcal{K}\}.$$

Changing prices changes the exposed face without changing the computation. Integer allocations, feasibility walls, and maxima of compute, memory, fabric, and bridge time require a more general finite phase object.

Definition 16.5 (Finite semialgebraic architecture registry). *Let $\Theta \subseteq \mathbb{R}^p$ be a semialgebraic technology domain. A finite registry consists of architecture classes $a \in \mathcal{A}$ and finite witness sets W_a . A witness $\omega \in W_a$ records every discrete realization, allocation, layout, and schedule choice. Its feasibility is described by finitely many polynomial or rational inequalities whose denominators have fixed sign on Θ ; its cost is assembled from rational functions by finitely many min and max operations. Infeasible witnesses have cost $+\infty$. Put*

$$C_a(\theta) = \min_{\omega \in W_a} C_{a\omega}(\theta), \quad \mathbf{A}(\theta) = \left\{ a : C_a(\theta) = \min_b C_b(\theta) < +\infty \right\}.$$

If every registered class is infeasible, $\mathbf{A}(\theta) = \emptyset$.

Theorem 16.6 (Finite technology-phase stratification). *Every registry satisfying Theorem 16.5 admits a finite partition of Θ into connected, relatively open semialgebraic sign cells, with relative openness taken in each cell's supporting stratum. On every cell the following complete set-valued labels are constant:*

- (i) feasible classes and witnesses;
- (ii) every active rational branch and bottleneck set, including ties;
- (iii) the optimal witness set within every class; and
- (iv) the globally winning architecture set $\mathbf{A}(\theta)$.

Every label change lies on a feasibility wall, branch or bottleneck exchange, witness exchange, architecture tie, or boundary of Θ .

Proof. Clear denominators on regions where their signs are fixed. Collect the finite family of feasibility polynomials, branch-comparison polynomials, within-class and cross-class cost differences, and defining polynomials of Θ . A sign-invariant semialgebraic cell decomposition exists for this finite family [46]; refine each stratum into connected cells. Every comparison then has one constant sign in $-, 0, +$. Retaining the zero sign preserves complete tied branch, bottleneck, witness, and architecture sets instead of inserting a hidden tie-break. The listed labels are consequently constant, and they can change only when a defining sign or domain stratum changes. $\square$

Corollary 16.7 (Certified full-label phase radius). *Let $\mathcal{P}$ contain every cleared feasibility polynomial, branch difference, within-class witness difference, and cross-class difference used in Theorem 16.6. Suppose $\theta_0 \in \Theta$, every $p \in \mathcal{P}$ is nonzero at θ_0 , and p is L_p -Lipschitz on $\overline{B}(\theta_0, r_0) \cap \Theta$. Then the complete phase label is unchanged throughout $B(\theta_0, r) \cap \Theta$ whenever*

$$r < \min \left\{ r_0, \min_{p \in \mathcal{P}: L_p > 0} \frac{|p(\theta_0)|}{L_p} \right\},$$

where the inner minimum is $+\infty$ if every $L_p = 0$.

Proof. For $|\theta - \theta_0|_2 < |p(\theta_0)|/L_p$, the Lipschitz inequality prevents p from reaching zero, so every defining sign is preserved. Intersecting with Θ also covers a point on the admissible-domain boundary. The radius controls losing-class witness exchanges and presently infeasible witnesses as well as the global winner. $\square$

A one-axis distance to a tie wall is therefore only a directional diagnostic; it is not the full-label radius unless every defining sign and the geometry of Θ have been included.

Computational complexity of exact phase and retention objects. The finite semialgebraic partition is an existence result, not a polynomial-time guarantee: general real quantifier elimination and cylindrical algebraic decomposition can be doubly exponential in the number of variables [114]. Likewise, finding a minimum-cardinality retained family that intersects every declared argmin set is a set-cover problem and is NP-hard in general [113]. Practical systems therefore use restricted parameterizations, incremental ledgers, certified bounds, and approximation algorithms while preserving certificate scope.

16.2 Certified inner and outer descriptions

Definition 16.8 (Certified outer relaxation). *For each hierarchy let*

$$\text{Adm}_\tau(\mathcal{T}) \subseteq \{(P, \phi) : P \in \mathfrak{R}_F^p(\mathcal{T}), \phi \in \text{Emb}_\tau(P, \mathcal{T})\}$$

be the exact admissible pairs. A lower-bounding relaxation consists of a superset

$$\text{Adm}_\tau^{\text{out}}(\mathcal{T}) \supseteq \{(\sigma_\tau(P), \phi) : (P, \phi) \in \text{Adm}_\tau(\mathcal{T})\}$$

and an optimistic objective satisfying

$$\Phi_{\tau, \mathcal{T}}^{\text{out}}(\sigma_\tau(P), \phi) \leq \Phi_\tau(P, \phi)$$

for every exact admissible pair.

Theorem 16.9 (Certified sandwich and safe pruning). *Let $\mathcal{J}_\in$ be a finite set of actual legal realization–embedding pairs. Then*

$$\boxed{\text{LB}_\tau \leq \text{OPT}_\tau^p(F) \leq \text{UB}_\tau,}$$

where

$$\text{LB}_\tau = \inf_{\mathcal{T}} \inf_{(z, \phi) \in \text{Adm}_\tau^{\text{out}}(\mathcal{T})} \Phi_{\tau, \mathcal{T}}^{\text{out}}(z, \phi), \quad \text{UB}_\tau = \min_{(P, \phi) \in \mathcal{J}_\in} \Phi_\tau(P, \phi).$$

A candidate class may be discarded whenever its scope-matched certified lower bound is strictly larger than the constructive incumbent.

Proof. Every exact admissible pair maps into the relaxed set with no larger relaxed cost, so the relaxed infimum is a lower bound. Every member of $\mathcal{J}_\in$ is an exact feasible design, so its best evaluated cost is an upper bound. A class whose lower bound exceeds that incumbent cannot contain an in-scope optimizer. $\square$

The scope check is part of the proof obligation. Computation, candidate class, objective and units, hierarchy, resource contract, technology contract, assumptions, and validity domain must match the decision being made. Any dependency change invalidates the affected certificate and reopens the corresponding branch. The retained object is the bracket and its dependency ledger, not merely the lowest evaluated point.

16.3 Lossless retention for a technology domain

Theorem 16.10 (Argmin-hitting retention). *Let $\mathcal{P}$ be a finite registered realization set and let Θ be any technology domain on which every technology has a finite attained optimum. Define*

$$\mathcal{M}_\theta = \underset{P \in \mathcal{P}}{\text{argmin}} C_\theta(P).$$

A retained family $\mathcal{A} \subseteq \mathcal{P}$ preserves the optimum at every technology in Θ if and only if

$$\boxed{\mathcal{A} \cap \mathcal{M}_\theta \neq \emptyset \quad \text{for every } \theta \in \Theta.}$$

A singleton is lossless exactly when $\bigcap_{\theta \in \Theta} \mathcal{M}_\theta \neq \emptyset$.

Proof. At one technology, the retained and full minima agree exactly when a retained realization attains the full minimum. Quantifying this equivalence over Θ proves both statements. $\square$

This is the exact statistic for a named technology domain. It is weaker than universal preservation for all continuous monotone objectives, which requires preserving the complete upper obligation region, and weaker than preservation for every additive nonnegative price, which requires preserving its convex upper body.

17 Architecture–technology scale and link matching

17.1 Scale-dependent communication strain

Let $R(g) > 0$ be the demand profile of one fixed realization and hierarchy, and let $B_\tau(g) > 0$ be the corresponding boundary-capacity profile of a technology. Their quotient

$$\chi_\tau(g) = \frac{R(g)}{B_\tau(g)}$$

measures scale-dependent strain.

Theorem 17.1 (Scaling strain). *If $R \in \text{RV}_p$ and $B_\tau \in \text{RV}_q$, then*

$$\chi_\tau \in \text{RV}_{p-q}.$$

Thus demand below the capacity index relaxes asymptotically, demand above it becomes increasingly mismatched, and equal indices are marginal, with constants and slowly varying factors deciding feasibility.

Proof. Write $R(g) = g^p \ell_R(g)$ and $B_\tau(g) = g^q \ell_B(g)$. Then $\chi_\tau(g) = g^{p-q} \ell_R(g)/\ell_B(g)$, and the quotient of positive slowly varying functions is slowly varying. $\square$

Corollary 17.2 (Homogeneous dimensional threshold). *In an ideal homogeneous d -dimensional substrate with boundary capacity $B_d(g) = \Theta(g^{(d-1)/d})$, a demand profile of index p has strain index*

$$p - \frac{d-1}{d}.$$

The marginal indices are $1/2$ in two dimensions and $2/3$ in three.

This is a scoped capacity statement, not an automatic routability theorem. Channel conflicts, pinning, timing, power delivery, design rules, and thermal constraints remain in Adm_τ . A fixed number of tiers changes only a constant; a tier count growing with problem size changes the model and must be registered explicitly.

17.2 Aligned links and chiplet interfaces

For one hierarchy-aligned realization, let t_e be required traffic per operation, b_e be payload per link transaction or cycle, and ϵ_e be link energy per bit.

Theorem 17.3 (Aligned-link feasibility and serialization). *In the independent fluid-link model, a one-shot embedding is feasible exactly when $t_e \leq b_e$ for every interface. The minimum uniform fluid and integral serialization factors are*

$$s^*(t, b) = \max_e \frac{t_e}{b_e}, \quad s_{\text{int}}^*(t, b) = \max_e \left\lceil \frac{t_e}{b_e} \right\rceil,$$

and the declared linear link energy is $E(t) = \sum_e \epsilon_e t_e$.

Proof. Each inequality is necessary by link capacity. It is jointly sufficient because the model declares independent links and no additional causal dependency. Uniform serialization requires $t_e \leq s b_e$ on every edge, which gives the maximum ratio and its integral ceiling. Energy adds by the declared per-bit accounting rule. $\square$

Now let one physical link unit provide payload rate B_e , b_e bits per cycle, energy ϵ_e per bit, and shoreline area a_e . Let f_{op} be the required operation rate and n_e the number of parallel link units.

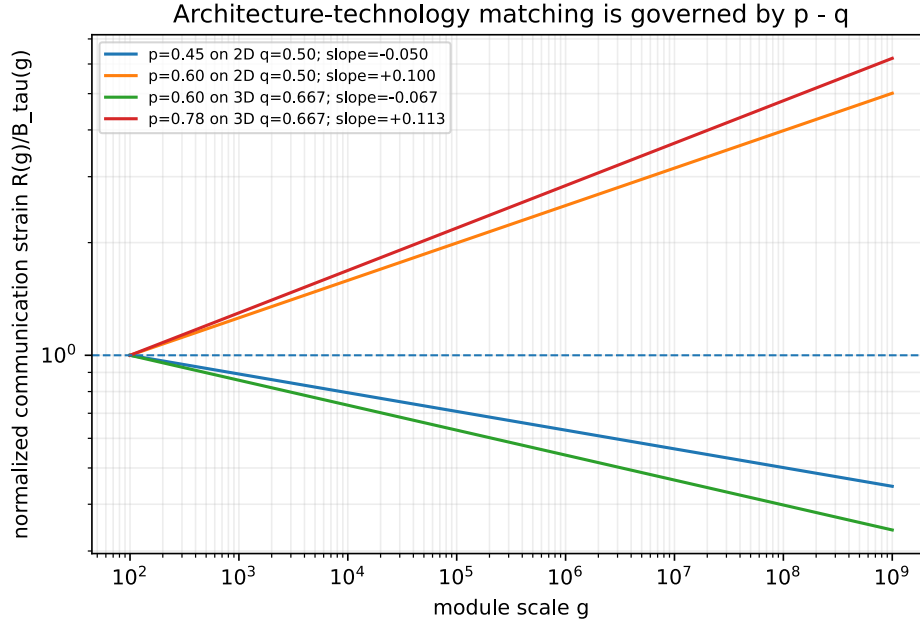


Figure 10: Normalized strain has log-log slope (p-q) in the declared model.

Theorem 17.4 (Chiplet link-budget law). *Every embedding of the selected realization satisfies*

$$\boxed{n_e \geq \left\lceil \frac{f_{\text{op}} t_e}{B_e} \right\rceil}, \quad \boxed{E_{e,\text{op}} \geq \epsilon_e t_e}, \quad \boxed{P_e \geq f_{\text{op}} \epsilon_e t_e},$$

$$\boxed{A_{\text{IO}} \geq \sum_e a_e n_e}, \quad \boxed{L_{e,\text{ser}} \geq \left\lceil \frac{t_e}{n_e b_e} \right\rceil}.$$

Any reach or shoreline limit with $n_e > n_e^{\max}$ certifies that the selected partition is infeasible under the declared package model.

Proof. The aggregate payload demand is $f_{\text{op}} t_e$, which gives the link-count bound. Per-bit accounting gives energy and power; link areas add. Parallel units carry at most $n_e b_e$ bits per cycle, yielding the serialization bound before propagation, protocol, buffering, and synchronization overhead. $\square$

The technology supplies $B_e, b_e, \epsilon_e, a_e$, reach, and protocol overheads; it does not change t_e unless the legal realization changes. UCie rate and package classes, for example, alter the embedding coordinates without erasing a semantic bridge [10].

18 Reference intermediate representation and compiler boundary

The calculus suggests a cross-layer loop with five roles:

1. a semantic front end registers the computation and legal changes to dataflow, tiling, storage, replication, approximation, ownership, and terminal meaning;
2. a realization generator emits constructive legal classes and preserves realization identity separately from typed obligation signatures;
3. a certificate engine combines semantic lower bounds, scoped outer relaxations, and constructive witnesses;
4. a technology library distinguishes price changes from capability changes that alter the resource contract or the legal realization family; and
5. physical tools return routed or otherwise evaluated constructions as new inner witnesses and can trigger targeted challengers when a bottleneck becomes active.

A reference implementation now exercises this boundary on a real PyTorch FX symbolic trace of a Transformer microblock. The prototype emits a computation contract, a resource contract, legal transformation classes, six constructive realization classes, typed obligations, three scoped outer bounds, and nine technology/objective decision certificates. It operationalizes the left half of the CRG abstraction ladder and the dependency record needed for safe repricing and reopening. It does not automatically enumerate every legal realization, generate general converses, ingest arbitrary compiler dialects or RTL, or perform physical signoff; those remain separate research and engineering problems.

OpenROAD demonstrates autonomous RTL-to-GDS exploration, and its ArtNet flow accepts hierarchical structure and scalar Rent parameters [42, 47, 48]. A production CRG adapter would exchange typed realization records, interface-separation distributions, certified inner/outer regions, decision margins, and dependency hashes instead of one scalar exponent. The operational rule is: analyze once and reprice many times only while legality, feasibility, and certificate scope remain unchanged. A capability change reopens affected branches and requires reconstruction rather than silent reuse of an obsolete geometry.

19 Analytic component certificates

The design calculus becomes operational when a computation has both a scoped converse and constructive realizations. Dense matrix multiplication and exact attention provide complementary cases. Matrix multiplication exposes a memory–traffic–collective frontier; attention exposes a traffic–storage–recomputation frontier. Neither has a technology-independent scalar winner.

19.1 Dense matrix multiplication across memory levels

Consider classical $n \times n$ dense matrix multiplication restricted to the conventional $\Theta(n^3)$ scalar-product DAG. For a two-level memory hierarchy with M fast words, established pebble and geometric bounds give $Q(M) = \Omega(n^3/\sqrt{M} + n^2)$, while blocked algorithms attain the same order under the usual cache assumptions [5, 6].

Theorem 19.1 (Multilevel matrix-multiplication bracket). *Let $M_1 < \dots < M_L$ be the cumulative fast-side capacities exposed by an inclusive multilevel contract, and let q_i be words moved across interface i . Inputs begin beyond each modeled interface and the final output is returned beyond it. There are constants $c_0, C_0 > 0$ such that every legal classical realization satisfies*

$$q_i \geq c_0 \left(\frac{n^3}{\sqrt{M_i}} + n^2 \right) \quad (1 \leq i \leq L),$$

while one cache-oblivious recursively blocked realization simultaneously satisfies

$$q_i \leq C_0 \left(\frac{n^3}{\sqrt{M_i}} + n^2 \right) \quad (1 \leq i \leq L)$$

at every level obeying the declared ideal-cache assumptions [7]. Thus one outer orthant and one jointly realizable inner point certify the multilevel traffic projection within a constant factor.

Proof. At level i , contract every faster level into the declared cumulative M_i -word fast side and apply the two-level lower bound to that complete interface. For achievability, the same cache-oblivious recursion induces a legal blocked schedule at every ideal-cache level simultaneously. The claim therefore does not combine incompatible per-level optima or count one transfer at several levels without accounting. A noninclusive hierarchy must replace M_i by the actual capacity available on the contracted fast side. $\square$

The same structured computation also yields a scale exponent. For a set $S \subseteq [n]^3$ of $g = |S|$ multiplication triples, the discrete Loomis–Whitney inequality gives

$$|S|^2 \leq |\pi_A(S)| |\pi_B(S)| |\pi_C(S)|.$$

Hence at least one operand/output projection has size $\Omega(g^{2/3})$; cubic blocks attain $O(g^{2/3})$. The natural classical blocked realization therefore has the interior demand law

$$R(g) = \Theta(g^{2/3}).$$

This is a result for the registered structured computation, not for every Boolean circuit computing matrix multiplication.

Table 6: Declared public technology-coordinate library. Payload rates are ideal one-direction cluster projections, not protocol or contention measurements.

Technology row	Payload	Link energy	Status
UCIe-S, 32 GT/s	64 GB/s	0.50 pJ/bit	public target anchor
UCIe-A, 32 GT/s	256 GB/s	0.25 pJ/bit	public target anchor
UCIe-A, 64 GT/s	512 GB/s	0.25 pJ/bit	rate anchor; energy proxy
Local fast state	32 or 96 MiB	–	public cache-capacity anchors

Table 7: Modeled $n = 8192$, $K_{\text{chip}} = 64$ projection with two-byte words. Time uses one ideal UCIe-A32 cluster on the critical interface.

c	Memory/chiplet	Traffic/chiplet	Stages	Transfer time	Link energy
1	6.0 MiB	32.00 MiB	8	131.1 μs	2.147 mJ
2	12.0 MiB	22.63 MiB	4	92.7 μs	1.519 mJ
4	24.0 MiB	16.00 MiB	3	65.5 μs	1.074 mJ

19.2 A certified 2.5D technology map

On K_{chip} processor-chiplets with per-chiplet fast memory M , classical distributed lower bounds include a per-chiplet communication floor of order $\Omega(n^3/(K_{\text{chip}}\sqrt{M}))$ under the declared distribution assumptions [8, 6]. The 2.5D family introduces a replication factor $1 \leq c \leq K_{\text{chip}}^{1/3}$ and, up to constants and logarithmic latency terms, obeys

$$M(c) = \Theta\left(\frac{3cn^2}{K_{\text{chip}}}\right), \quad Q(c) = \Theta\left(\frac{n^2}{\sqrt{cK}}\right), \quad R(c) = O\left(\sqrt{\frac{K_{\text{chip}}}{c^3}} + \log c\right)$$

[9]. The numerical certificate declares the constructive leading model

$$q(c) = \frac{2n^2}{\sqrt{cK}}, \quad m(c) = \frac{3cn^2}{K_{\text{chip}}}, \quad r(c) = \left\lceil \sqrt{K_{\text{chip}}/c^3} \right\rceil + \lceil \log_2 c \rceil.$$

The constants are model coordinates; the exchange of memory for traffic and collective stages is inherited from the cited algorithms and converses.

For $K_{\text{chip}} = 64$, two-byte words, and $c \in \{1, 2, 4\}$, feasibility under local memory budget M_{max} is exactly

$$\frac{3cn^2}{K_{\text{chip}}}(2 \text{ bytes}) \leq M_{\text{max}}.$$

Both (qc) and (rc) decrease with c , so the largest feasible replication factor is Pareto dominant for every monotone traffic-round objective in this declared family.

The UCIe rows use public rate classes and the declared energy proxies [10, 11]; the local-state anchors follow public processor documentation [12]. Together they define a public-anchor technology library rather than the specification of one shipping package.

The resulting memory-conditioned selection is

32 MiB/chiplet	c^*
$n \leq 9459$	4
$9459 < n \leq 13377$	2
$13377 < n \leq 18918$	1
$n > 18918$	infeasible in the declared local-memory model.

With 96 MiB per chiplet, the $c = 4$ region extends to $n = 16384$. The public link rows quantify the value of a selected realization but do not change the largest-feasible- c ordering in this idealized family.

Collective capability can reverse a different pair of realizations. At $K_{\text{chip}} = 64$, a one-dimensional replicated- B point with effective collective stage count r has typed signature

$$z_{1D}(r) = (63, r, 1.03125),$$

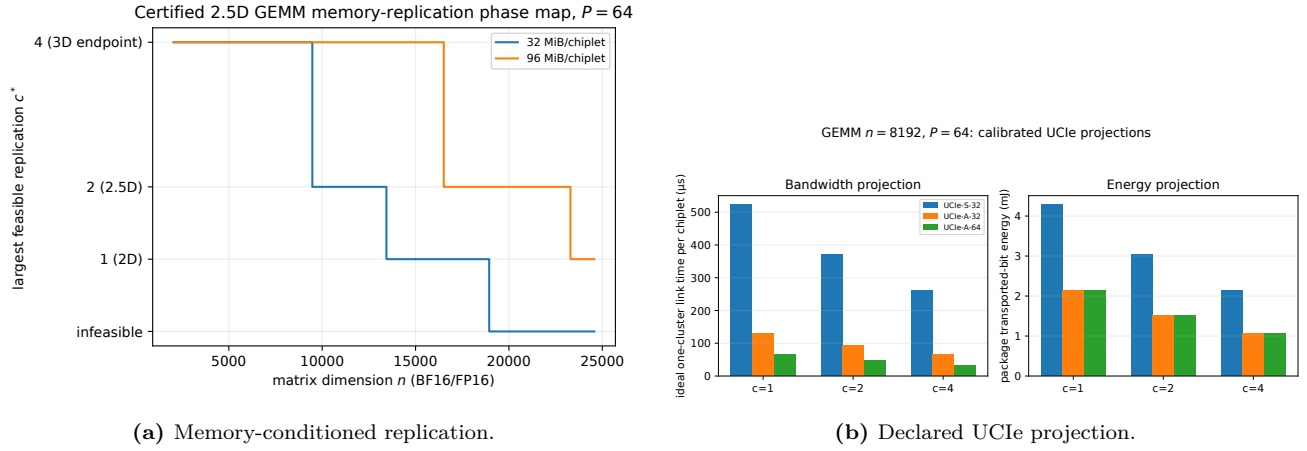


Figure 11: The 2.5D realization family is retained until memory and link technology are declared.

while the declared two-dimensional SUMMA point is $z_{2D} = (14, 8, 0.046875)$ [13, 14]. Under price $C = w_Q q + w_R r + w_M m$, the one-dimensional realization can win only for $r < 8$, and then exactly when

$$w_R > \frac{49w_Q + 0.984375w_M}{8 - r}.$$

The reversal is attached to an explicit collective capability rather than an unstated communication primitive.

19.3 Exact attention: recomputation versus storage

Let $Q, K, V \in \mathbb{R}^{N \times d}$ and consider exact attention $O = \text{softmax}(QK^T)V$ with fast memory M scalar elements. The conventional realization materializes $N \times N$ score and probability matrices; the tiled realization retains rowwise normalization state and recomputes blocks when required.

Theorem 19.2 (Imported exact-attention I/O bracket). *For $d \leq M \leq Nd$, standard exact attention has*

$$Q_{\text{std}} = \Theta(Nd + N^2), \quad S_{\text{std}} = \Theta(N^2),$$

whereas *FlashAttention achieves*

$$Q_{\text{flash}} = \Theta\left(\frac{N^2 d^2}{M}\right), \quad S_{\text{flash}} = O(N),$$

while spending additional local work through recomputation. The cited analysis proves I/O optimality over the stated SRAM range [15].

The theorem supplies incomparable typed points, not one scalar winner. For $d = 64$ and 128, the asymptotic crossover $M > d^2$ corresponds to only 8 and 32 KiB for two-byte elements, but implementation constants and additional work still require an explicit objective.

For the published A100 forward-plus-backward GPT-2-medium measurement, the reported points are

$$z_{\text{std}} = (Q, F, T) = (40.3 \text{ GB}, 66.6 \text{ GFLOP}, 41.7 \text{ ms}),$$

$$z_{\text{flash}} = (4.4 \text{ GB}, 75.2 \text{ GFLOP}, 7.3 \text{ ms})$$

[15]. Under typed additive cost $C = w_Q Q + w_F F + w_S S$, the tiled realization is preferred exactly when

$$35.9w_Q + \Delta S w_S > 8.6w_F, \quad \Delta S = S_{\text{std}} - S_{\text{flash}} > 0.$$

With zero storage price, the break-even is approximately 0.240 FLOP/byte. Public A100 peak/bandwidth ratios are approximately $838\times$ larger on the BF16 tensor-peak basis and about $52\times$ larger on the FP32 basis [16]; this establishes the arithmetic-intensity separation; measured kernel time additionally depends on utilization, scheduling, and the remaining typed coordinates.

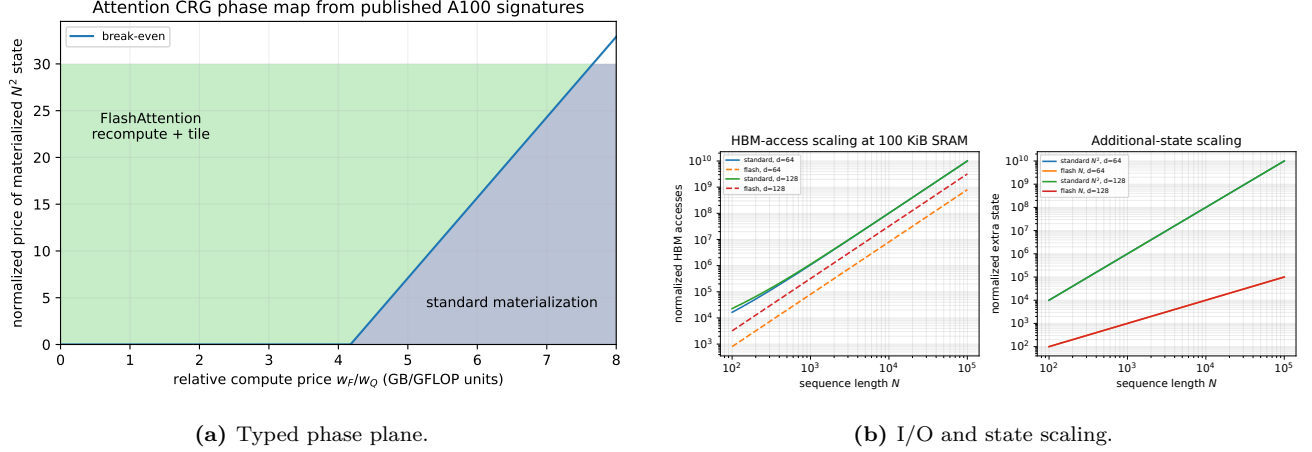


Figure 12: Traffic, work, and storage jointly determine the attention realization boundary.

19.4 Cross-stage state and layout laws

The end-to-end application joins attention and an MLP. Two general facts prevent isolated stage optima from being multiplied together without a joint resource check.

Theorem 19.3 (Static provisioning and time-shared state). *Let two isolated modes require usable state m_A and m_F on a chiplet with budget M .*

- (i) *A static fused realization with dedicated simultaneously instantiated engines is infeasible when $m_A + m_F > M$.*
- (ii) *A sequential universal realization that reuses one state pool is state-feasible when $\max(m_A, m_F) \leq M$.*

The two statements belong to different realization contracts; time sharing must expose any efficiency cost rather than treating reconfiguration as free.

Proof. Dedicated state occupies disjoint capacity and adds. State used in disjoint time intervals occupies the maximum. An embedding cannot transform one resource aggregator into the other without changing the realization. $\square$

For layout conversion, let $K_{\text{chip}} = CT = C'T'$ chiplets implement aligned two-dimensional layouts over a normalized activation tensor. Let $S_{i=1}^{K_{\text{chip}}}$ and $T_{j=1}^{K_{\text{chip}}}$ be the equal-area source and target rectangles. Each tensor element has one source and one target owner; chiplet relabeling is free and bijective. Compression, recomputation, replication, coding, cached duplicates, unequal-capacity placement, routed congestion, latency, and energy are outside this conversion contract.

Theorem 19.4 (Exact aligned-layout assignment). *After optimal chiplet relabeling, the exact retained and minimum moved fractions are*

$$r^*(C, C') = \max_{\pi \in \mathfrak{S}_{K_{\text{chip}}}} \sum_{i=1}^{K_{\text{chip}}} |S_i \cap T_{\pi(i)}|, \quad \mu^*(C, C') = 1 - r^*(C, C').$$

If $C \mid C'$ or $C' \mid C$, then

$$r^*(C, C') = \frac{\min(C, C')}{\max(C, C')}, \quad \mu^*(C, C') = 1 - \frac{\min(C, C')}{\max(C, C')}.$$

For nonnested factors the closed form need not hold.

Proof. Under a relabeling π , chiplet i retains exactly $|S_i \cap T_{\pi(i)}|$, so maximizing retention is a maximum-weight perfect assignment. Retain every selected intersection and move only each target remainder from its unique source owner; this realizes movement equal to one minus retained measure. Conversely, every element outside the retained intersections must change owner.

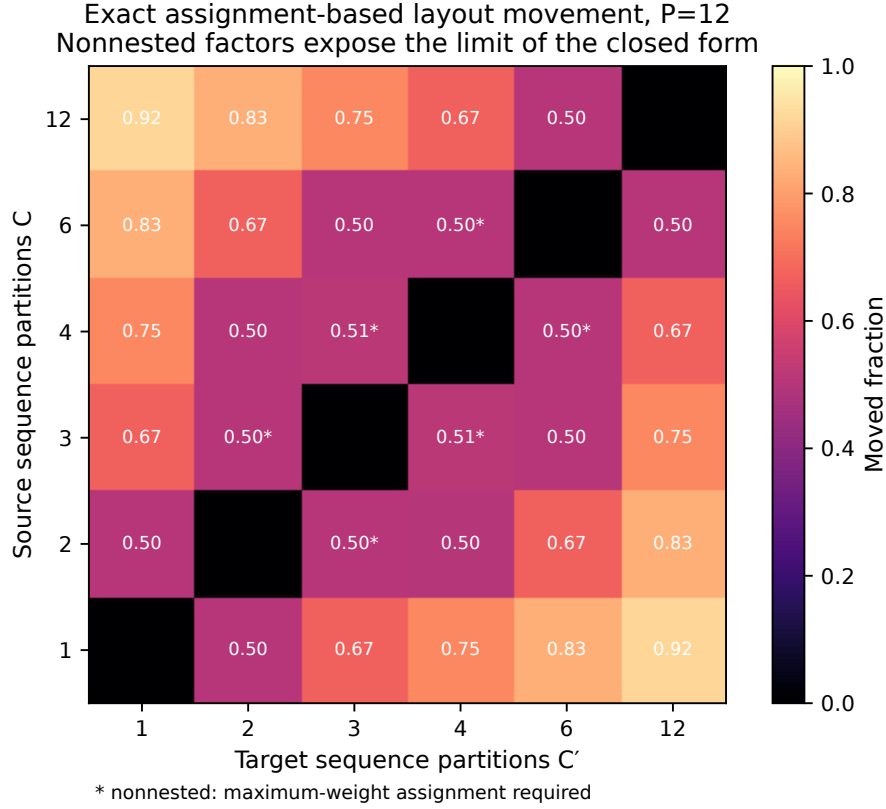


Figure 13: The $K_{\text{chip}} = 12$ nonnested example separates the general assignment law from the nested closed form.

For the nested case $C' = kC$, each nonempty intersection has area at most $1/(kK)$, so total retention is at most $1/k$. The intersection graph is k -regular and bipartite, hence has a perfect matching attaining $1/k$. The case $C > C'$ is symmetric. $\square$

At $K_{\text{chip}} = 12, C = 4, C' = 6$, the exact retained fraction is $1/2$, not $2/3$, which shows why nesting is load bearing. Auto-sharding systems already price resharding; the result here is the exact optimal-relabeling law under its declared ownership contract [17, 18].

These component certificates now supply the end-to-end case with scoped converses, constructive stage modes, state aggregators, and an exact ownership conversion law. The remaining question is which joint realization wins once all resources and package obligations are conserved simultaneously.

Part IV

End-to-end certificate

20 Certified nonseparable co-design of a long-context Transformer superblock

The component certificates become decision-relevant only after dataflow, state, chiplet ownership, memory allocation, execution semantics, and package service are registered jointly. The case study considers one exact residual superblock

$$Y = X + \text{GQA}(X), \quad Z = Y + \text{SwiGLU}(Y), \quad (20.1)$$

for batch-one prefill. The conditional question is:

Given a declared workload-execution contract, a finite architecture registry, and a technology ledger, which registered class minimizes the declared steady-state initiation interval?

The answer is exhaustive for the declared finite algebraic model. The model contains the registered architecture classes, execution policies, resource aggregators, ownership obligations, integer allocations, and technology coordinates needed to derive the class decision and its phase structure.

20.1 Workload and execution contract

The primary instance uses hidden dimension $d = 4096$, feed-forward width $d_{\text{ff}} = 14336$, $h = 32$ query heads, $h_{\text{KV}} = 8$ key/value heads, two-byte BF16/FP16 elements, and contexts through $N = 131,072$ [57, 58]. A multiply-add counts as two FLOPs. If $M_{\text{exec}}(N)$ is the number of query-key pairs actually executed, the registered tensor-matmul work is

$$F_A(N) = \left(4 + 4 \frac{h_{\text{KV}}}{h}\right) N d^2 + 4 d M_{\text{exec}}(N), \quad (20.2)$$

$$F_F(N) = 6 N d d_{\text{ff}}. \quad (20.3)$$

The projection coefficient is five for the declared GQA ratio, not the MHA coefficient eight.

The three executable pair policies are

$$M_{\text{exec}}(N) = \begin{cases} N^2, & \text{dense-padded causal execution,} \\ N(N+1)/2, & \text{ideal causal-pair execution,} \\ B^2 m(m+1)/2, \quad m = \lceil N/B \rceil, & \text{block-rounded causal execution.} \end{cases} \quad (20.4)$$

The named default uses $B = 128$ and executes complete square tiles on and below the block-causal diagonal, including final-tile padding.

At $N = 16,384$, the semantic causal count is 134,225,920, the executed block-rounded count is 135,266,304, attention tensor work is 3.590593 TFLOP, and MLP tensor work is 5.772436 TFLOP. The equality $F_A = F_F$ at $N = 4d$ holds for these dimensions under dense-padded execution; it is not invariant to execution policy.

Softmax, RoPE, RMSNorm, residual additions, SwiGLU elementwise work, launch and control, synchronization, ECC, and unmodeled utilization losses are typed exclusions. They are not silently absorbed into Equations (20.2) and (20.3). Every timing claim is therefore scoped to the named tensor-work model.

One hidden-state tensor occupies

$$A(N) = N d b_{\text{elem}}, \quad b_{\text{elem}} = 2 \text{ bytes.}$$

If attention and MLP have disjoint single-owner regions, steady-state alternation moves one state to the MLP and one back to the next attention stage:

$$\boxed{Q_B(N) \geq 2 N d b_{\text{elem}}.} \quad (20.5)$$

At $N = 16,384$, $A = 128$ MiB and $Q_B = 256$ MiB. Placement can price this obligation but cannot erase it while disjoint ownership remains part of the contract.

20.2 Evidence-tiered technology contract

The declared system has $K_{\text{chip}} = 32$ compute chiplets, eight HBM placements, 8 TB/s effective aggregate HBM bandwidth, and 32 MiB usable fast state per planar fused chiplet. “Usable” is after declared software, fragmentation, and reliability derating; a witness exactly on a capacity wall has zero capacity margin.

The default dense BF16 tensor-work coordinate is 5 PFLOP/s. The corresponding 10-PFLOP/s public headline is a structured-sparse coordinate; it appears only in explicitly named sparse-peak sensitivity rows [59]. Peak is a ceiling, not delivered kernel throughput. The 8-TB/s HBM coordinate is a declared derating of public HBM capability rather than a specification for a shipping package [60].

Table 8: Declared package coordinates. Payload and energy are model coordinates anchored to public capability classes, not guarantees.

Technology	Payload	Link energy	Evidence status
Standard declared cluster	64 GB/s	0.50 pJ/bit	public-rate anchor
Advanced 32-GT/s cluster	256 GB/s	0.25 pJ/bit	declared sensitivity
Advanced 64-GT/s cluster	512 GB/s	0.25 pJ/bit	rate anchor; energy proxy
Vertical two-tier	4 TB/s	0.05 pJ/bit	declared capability study

UCIe supplies the public rate and 3D capability classes used as anchors [10]; the model does not infer aggregate payload, latency, or energy directly from raw lane rate. In the default *shared aggregate* contract, internal attention traffic, internal MLP traffic, and Equation (20.5) share one package-payload budget. A named sensitivity charges internal collectives to separate local fabrics and only the bridge to the package. The distinction is encoded in the contract name

llama31-gqa-block-rounded-causal128-dense-bf16-shared-package-aggregate-v1.

20.3 Finite architecture and mode registry

Each registered realization retains the typed signature

$$\sigma(P) = (F_{\text{loc}}, M_{\text{fast}}, Q_{\text{HBM}}, Q_{\text{pkg}}, R_{\text{coll}}, D_{\text{causal}}, E_{\text{link}}, T_{\text{II}}, T_{\text{lat}}, \Pi, \eta).$$

Capacity, traffic, bandwidth, latency, and reconfigurability remain distinct coordinates. The six architecture classes are:

H: static spatial fused.

Dedicated attention and MLP engines coexist; the stage bridge is local, but nonshareable state adds.

U: universal time-shared fused.

One reconfigurable fabric executes the stages sequentially, reuses one state pool, and pays universal attention efficiency η_U .

S: heterogeneous split.

Attention and MLP own disjoint compute and HBM pools and overlap in steady state; their internal traffic and semantic bridge consume the declared package resources.

X: dual-mode hybrid.

Split hardware also supports a universal fallback and selects the better legal U/S parent with a declared two-percent whole-path capability tax. At zero tax it ties, but cannot beat, the parent envelope without a new schedule.

M: memory-centric attention.

Attention uses a declared near-HBM, lower-compute-density capability; the MLP remains compute dense.

V: vertically specialized.

Two active tiers use twice the planar compute density, reduced internal traffic, and the declared vertical fabric.

Attention has an 8-MiB Flash-local mode and a 28-MiB collective-aware mode. The latter imports FlatAttention’s 0.893 utilization, $4.1\times$ relative speed, and $16\times$ HBM-traffic comparison as a portability hypothesis from a different MHA many-PE architecture [61]. The MLP has an 8-MiB low-replication mode and a 24-MiB $c = 4$ communication-avoiding mode. These are registered, target-specific mode coordinates within the declared realization contract.

The base prefill traffic ledger is

$$\begin{aligned}
W_A &= \left(2 + 2\frac{h_{KV}}{h}\right) d^2 b_{\text{elem}}, & W_F &= 3dd_{\text{ff}}b_{\text{elem}}, \\
Q_A^{(0)} &= W_A + 4A(N), & Q_F^{(0)} &= W_F + 6A(N), \\
D_A^{(0)} &= 2A(N), & D_F^{(0)} &= 2A(N).
\end{aligned} \tag{20.6}$$

The effective HBM and local-fabric bandwidth scales are 8 and 4 TB/s.

Table 9: Base stage-mode coordinates. Multipliers apply to Equation (20.6).

Mode	State	η	$Q/Q^{(0)}$	$D/D^{(0)}$	Peak scale
Attention Flash-local	8 MiB	0.893/4.1	16	1	1
Attention Flat-collective	28 MiB	0.893	1	0.5	1
MLP 2D-low-replication	8 MiB	0.85	1	1	1
MLP 2.5D-c4	24 MiB	0.85	0.5	0.5	1

H enumerates mode pairs whose state sum is at most 32 MiB. U enumerates pairs whose state maximum is at most 32 MiB and substitutes $\eta_U = 0.80$ for universal collective-aware attention. Thus Theorem 19.3 gives

$$28 + 24 = 52 > 32, \quad \max(28, 24) = 28 \leq 32.$$

The high-high pair is illegal in H but legal in U; U’s competitiveness is governed by its efficiency, not by a false sum of sequential state.

S fixes the collective-aware attention and $c = 4$ MLP modes and enumerates all positive integer partitions

$$(k_A, k_F) = (1, 31), \dots, (31, 1), \quad (h_A, h_F) = (1, 7), \dots, (7, 1).$$

There are $31 \times 7 = 217$ witnesses. X applies

$$T_X = \frac{\min(T_U, T_S)}{1 - 0.02}.$$

M fixes compute/HBM partitions $8/24$ and $5/3$ with a declared near-HBM attention mode. V uses the full 217-witness enumeration with twofold peak scale, half HBM traffic, one-tenth internal traffic, and the vertical package row.

When two stage modes use different aligned layouts, the exact represented-byte movement is the assignment value in Theorem 19.4; its nested closed form is used only when the partition factors are actually nested. The split class additionally carries the two-way bridge in Equation (20.5). No placement or relabeling is credited with eliminating bytes that the ownership contract requires to move.

20.4 Conserved timing and energy

For stage $j \in \{A, F\}$, let F_j, Q_j, D_j denote tensor work, HBM bytes, and internal-fabric bytes. With compute share α_j , HBM share β_j , utilization η_j , peak C_{pk} , HBM bandwidth B_{HBM} , and local fabric B_f , define

$$t_j^{\text{cmp}} = \frac{F_j}{\alpha_j C_{\text{pk}} \eta_j}, \quad t_j^{\text{HBM}} = \frac{Q_j}{\beta_j B_{\text{HBM}}}, \quad t_j^{\text{fab}} = \frac{D_j}{B_f}.$$

U reuses full resources sequentially:

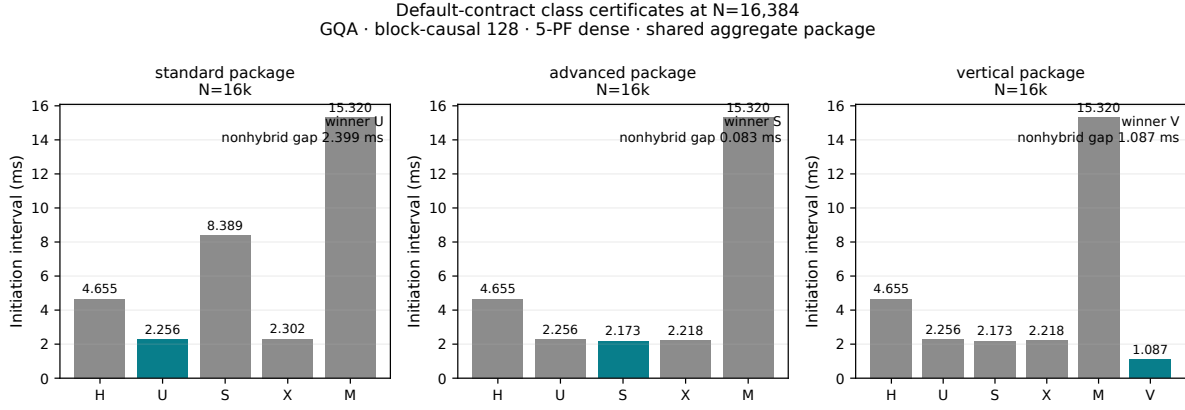
$$T_U = \max(t_A^{\text{cmp}}, t_A^{\text{HBM}}, t_A^{\text{fab}}) + \max(t_F^{\text{cmp}}, t_F^{\text{HBM}}, t_F^{\text{fab}}).$$

H is likewise registered with a sequential full-resource schedule. A hypothetical continuously overlapped stream cannot give both stages the full capacity: for full-resource demands a_r, f_r , conservation gives

$$\min_{\gamma_r \in (0,1)} \max_r \left\{ \frac{a_r}{\gamma_r}, \frac{f_r}{1 - \gamma_r} \right\} = \max_r (a_r + f_r). \tag{20.7}$$

Table 10: Default-contract class optima at $N = 16,384$, in milliseconds. X includes its two-percent tax.

Scenario	H	U	S	X	M	V	Winner / full gap
Standard 64 GB/s	4.655293	2.255868	8.388608	2.301907	15.319862	–	U / 0.046038
Advanced 512 GB/s	4.655293	2.255868	2.173152	2.217502	15.319862	–	S / 0.044350
Vertical 4 TB/s	4.655293	2.255868	2.173152	2.217502	15.319862	1.086576	V / 1.086576

**Figure 14:** Strict registered-class decisions. Thin X gaps are the declared hybrid tax; U/S parent gaps are reported separately.

Indeed each resource requires at least $a_r + f_r$, and choosing $\gamma_r = a_r / (a_r + f_r)$ attains that resource's bound. A continuous envelope is not promoted to a class witness without a common legal integer schedule.

For each S witness under shared-package accounting,

$$T_S = \max \left\{ t_A^{\text{cmp}}, t_A^{\text{HBM}}, t_F^{\text{cmp}}, t_F^{\text{HBM}}, \frac{D_A + D_F + Q_B}{B_{\text{pkg}}} \right\}. \quad (20.8)$$

Under separate-local-fabric accounting, the final term is Q_B / B_{pkg} and D_A, D_F are charged to their divided local-fabric pools. The reported S value is the exact minimum over all 217 integer witnesses. Initiation interval is not conflated with split pipeline-fill latency, which is reported separately. Package link energy is the typed coordinate

$$E_{\text{pkg}} = 8Q_{\text{pkg}}\epsilon_{\text{bit}};$$

this is the package-link energy coordinate; full-system energy is represented by additional signature coordinates.

Model precision and physical fidelity. Exact decimal values below identify the registered algebraic model and support reproducibility. The declared auxiliary-kernel exclusions can shift wall locations by amounts comparable to the printed margins. In adversarial costing, the U/S class ordering remained stable while the efficiency wall moved from about 0.881 into roughly 0.847–0.872. Prose therefore reports wall locations to about three significant figures; exact machine values remain in the released records.

20.5 Strict registered decisions and phase walls

The standard package selects U at 2.255868 ms. The advanced package selects S at 2.173152 ms, and the vertical capability selects V at 1.086576 ms. The advanced S witness has unique compute split 12/20; all 7 positive HBM partitions tie in initiation interval, with 1/7 used only as a deterministic representative. The semantic bridge takes 0.524288 ms at 512 GB/s. Under shared-package accounting, internal traffic plus the bridge totals 512 MiB per interval and 1.073742 mJ at 0.25 pJ/bit. The separately reported fill latency is 4.841878 ms.

At the advanced point, the U/S efficiency wall is

$$\boxed{\eta_U^* = 0.881200.} \quad (20.9)$$

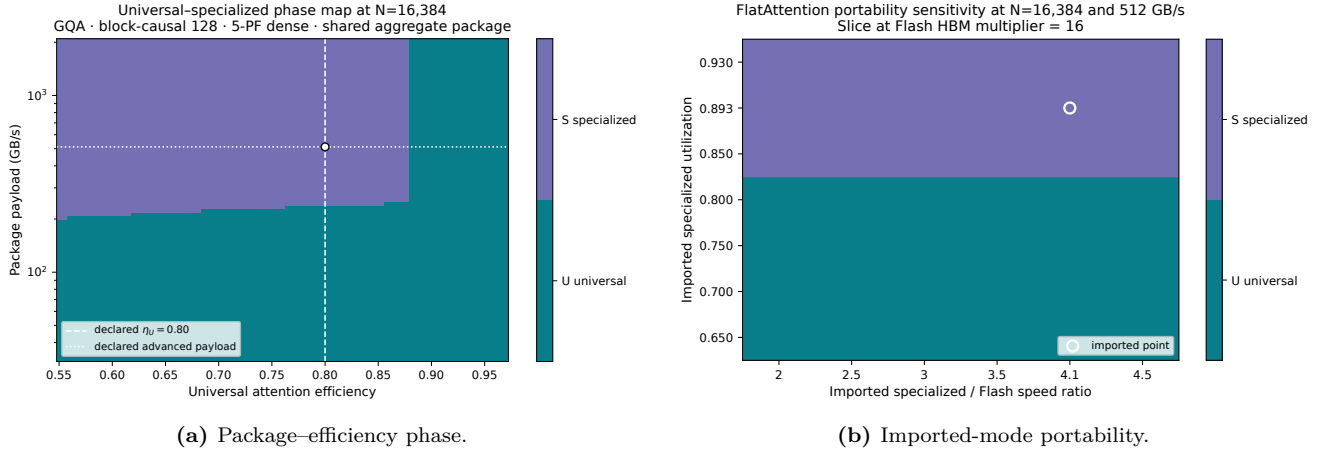


Figure 15: The advanced S point is strict but boundary-proximal.

The declared $\eta_U = 0.80$ is only 0.081200 below that wall. Holding other coordinates fixed, the U/S package-payload wall is 237.988577 GB/s. These unlike coordinate distances are reported separately. They are directional diagnostics, not the full-label Euclidean certificate of [Theorem 16.7](#).

The FlatAttention portability grid has 210 combinations: seven specialized utilizations, six speed ratios, and five Flash HBM multipliers. U wins 120 and S wins 90. The imported 0.893, 4.1 $\times$, 16 $\times$ point is a registered portability coordinate; the declared GQA realization determines its target-specific value.

The declared finite registry meets [Theorem 16.5](#): every mode and integer allocation is finite, and its timing laws are rational terms joined by finite minima and maxima. The winner, active bottleneck, and optimal witness sets are therefore piecewise constant on the semialgebraic phase cells of [Theorem 16.6](#).

20.6 Context scaling and twelve-contract sensitivity

The tensor-work profile has the form

$$R_{\text{block}}(N) = 4dM_{\text{exec}}(N) + \left[\left(4 + 4\frac{h_{KV}}{h} \right) d^2 + 6dd_{\text{ff}} \right] N.$$

For a smooth dense-padded specialization $aN^2 + \beta N$, its local exponent is

$$p_{\text{loc}}(N) = \frac{2aN^2 + \beta N}{aN^2 + \beta N},$$

which moves from one toward two. The bridge is an exponent-one mode. Its fraction decreases asymptotically, but its finite-scale affordability depends on package bandwidth, fast-state feasibility, and coefficients; exponents alone do not select the class.

Under the default contract U is strictly faster for $1 \leq N \leq 131,072$. In the exhaustive integer enumeration through 1,048,576, S is no slower on

$$[198,913, 198,926] \quad \text{and} \quad [199,041, 1,048,576].$$

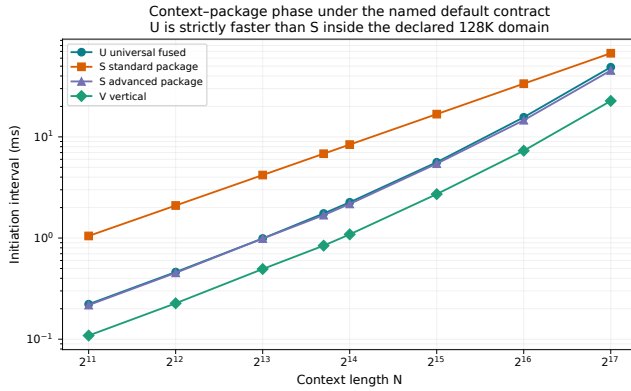
U retakes the lead on [198,927, 199,040], so “sustained” refers only to the finite evaluated range and makes no claim beyond it.

The U/S/V reversal at $N = 16,384$ holds in every evaluated row, but exact walls do not: η_U^* spans 0.85 to 0.8812004, and standard-package dominance transitions move inside or beyond the declared domain. Every block-causal row has two S-no-slower intervals because tile rounding makes the predicate nonmonotone on the finite range. The machine-readable result records retain every interval rather than reporting only one crossover.

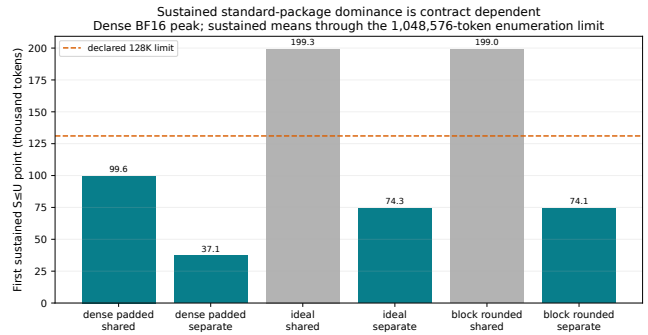
For the three primary technology scenarios, the strict argmin classes are U, S, and V. Their intersection is empty, so [Theorem 16.10](#) rules out every lossless singleton across that ledger. The family $\{U, S, V\}$ hits every argmin set and has zero class-level regret. X is not an argmin in a primary row at its declared tax; at zero tax it can hit U/S but not V. This is the exact retained-family result for the declared three-scenario technology ledger.

Table 11: Twelve-contract U/S sensitivity. The peak is either 5-PF dense or 10-PF structured-sparse sensitivity. A dagger marks an event outside the declared 128K domain.

Executed pairs / peak	Package accounting	η_U^*	Earliest	Sustained
Dense padded / 5 PF	shared aggregate	0.850000	99,641	99,641
Dense padded / 5 PF	separate local	0.850000	37,141	37,141
Dense padded / 10 PF	shared aggregate	0.850000	224,641†	224,641†
Dense padded / 10 PF	separate local	0.850000	99,641	99,641
Ideal causal / 5 PF	shared aggregate	0.877017	199,281†	199,281†
Ideal causal / 5 PF	separate local	0.877017	74,281	74,281
Ideal causal / 10 PF	shared aggregate	0.877017	449,281†	449,281†
Ideal causal / 10 PF	separate local	0.877017	199,281†	199,281†
Block causal-128 / 5 PF	shared aggregate (default)	0.881200	198,913†	199,041†
Block causal-128 / 5 PF	separate local	0.881200	73,985	74,113
Block causal-128 / 10 PF	shared aggregate	0.881200	449,025†	449,153†
Block causal-128 / 10 PF	separate local	0.881200	198,913†	199,041†



(a) Default context/package phase.



(b) Contract-sensitive transition.

Figure 16: The default has no standard-package S overtake inside 128K; gray events are finite outside-domain diagnostics.

20.7 Evidence ledger and falsification conditions

Table 12: Claim–evidence separation for the end-to-end certificate.

Class	Contents
Exact identities	GQA tensor work; pair policies; semantic bridge; layout assignment; state sum/max; conserved resources; complete integer enumeration
Imported evidence	Public model dimensions and context; FlashAttention I/O analysis; FlatAttention MHA ratios; UCle, HBM, and accelerator specifications
Declared coordinates	Usable fast state; delivered utilization; aggregate package payload and energy; universal efficiency; vertical capability; hybrid tax
Constructive witnesses	Registered schedules, allocations, initiation intervals, fill latency, package traffic, and link energy
Derived certificates	Winners, margins, phase walls, portability sweep, context intervals, and finite-ledger retention result
Unresolved calibration	Executed kernel trace; achieved GQA efficiency; fabric topology; auxiliary work; routed area/timing/power; thermal, yield, and cost

All technology coordinates are public, synthetic, or explicitly declared; none is presented as customer, supplier, production, or proprietary process data. The primary certificate is withdrawn, weakened, or reclassified if:

- a kernel trace selects a different execution policy and the matching sensitivity changes the winner;
- the declared GQA target reaches the structured-sparse peak, or fails to approach the dense-mode efficiencies, without updating the contract;
- internal collective traffic and the bridge do not share the declared aggregate package resource;
- target measurements invalidate imported attention state, utilization, speed, or traffic coordinates;

- credible derating crosses the U/S wall or a fast-state feasibility wall;
- an omitted legal class or genuinely new hybrid schedule beats the registered envelope; or
- the objective changes from steady-state initiation interval to latency, energy, area, cost, thermal behavior, or another quantity that selects a different class.

The executable model reproduces the workload equations, mode and technology ledgers, all 217 S and V allocations, twelve-contract sweep, complete context intervals, assignment oracle, portability grid, tables, and figures. Unit tests check arithmetic identities, conservation, phase decisions, and artifact integrity. These checks establish reproducibility of the complete finite model and bind each reported decision to its equations, registry, witness enumeration, and artifact record.

The case closes the manuscript’s design chain. Stagewise-optimal modes can be jointly infeasible; time sharing restores state feasibility at an explicit efficiency price; splitting restores specialization while activating a semantic bridge; and technology changes which legal realization wins. A downstream optimizer can improve the embedding of a committed graph, but it cannot recover a realization that upstream commitment removed.

Part V

Executable realization and end-to-end validation

21 From a compiler graph to physical consequence

The formal objects of CRG admit a direct executable realization. A traced computation graph provides operator semantics, tensor dependencies, shapes, and terminal ownership. A resource contract declares legal transformations. Transformation instances generate constructive realizations. A signature extractor maps each realization to typed joint obligations. Converse modules contribute scoped outer obligations. Technology models then select and physically evaluate realizations. The reference implementation carries this chain from an exact Transformer microblock to two routed heterogeneous-package models:

$$G_{\text{FX}} \longrightarrow (F, \rho, \mathcal{T}) \longrightarrow \mathcal{P}_{\text{ref}} \xrightarrow{\sigma} \mathcal{R}_{\text{ref}} \longrightarrow \text{safe reduction} \longrightarrow \text{technology embedding.} \quad (21.1)$$

21.1 Reference CRG intermediate representation

Let G_{FX} denote the PyTorch FX symbolic trace of a microblock containing query, key, and value projections, score formation, softmax, attention-value multiplication, output projection, residual connections, feed-forward expansion and contraction, nonlinear activation, and final residual integration. The trace is converted into

$$\text{IR}_{\text{CRG}} = (F, \rho, \mathcal{T}, \mathfrak{T}, \mathcal{P}_{\text{ref}}, \sigma, \mathcal{B}_{\text{out}}), \quad (21.2)$$

where F is the terminalized computation, ρ the resource contract, $\mathcal{T}$ the ownership hierarchy, $\mathfrak{T}$ the transformation grammar, $\mathcal{P}_{\text{ref}}$ the constructive realization family, σ the typed signature extractor, and $\mathcal{B}_{\text{out}}$ the set of class-scoped outer obligations.

The computation contract records operators, tensor shapes, input/output ownership, element type, exactness, and the correspondence between graph values and terminal data. The resource contract records materialization, recomputation, fusion, split ownership, replication, time sharing, and vertical placement as legal transformation classes. Applying that grammar generates six realizations:

1. fused materialization;
2. fused recomputation;
3. split materialization;
4. split recomputation;
5. replicated split recomputation; and
6. vertical split recomputation.

For $P \in \mathcal{P}_{\text{ref}}$, the extractor emits

$$\sigma(P) = (W, M_{\text{fast}}, Q_{\text{HBM}}, Q_{\text{pkg}}, Q_{\text{bridge}}, D, L, E, A, T_{\text{max}}, Y_{\text{proxy}}, R_{\text{proxy}})(P), \quad (21.3)$$

with a unit, aggregation law, and evidence source attached to every coordinate. Three outer-obligation records constrain semantic bridge movement, state capacity, and service in their registered classes. Combining the

Reference CRG intermediate representation

A working bridge from compiler graph to proof-carrying decision

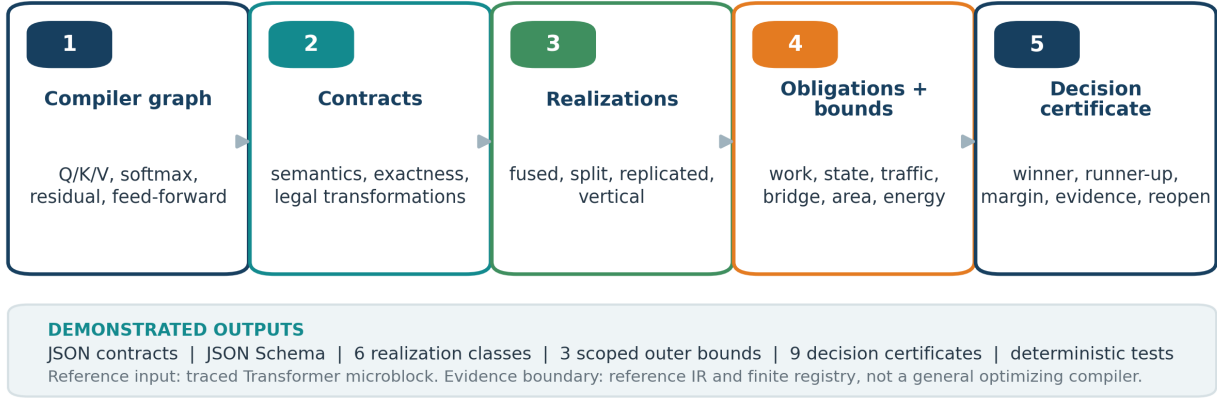


Figure 17: Reference graph-to-certificate pipeline. A traced compiler graph is expanded into computation and resource contracts, six constructive realizations, typed obligations, scoped outer bounds, and technology-conditioned decision certificates.

realization family, signatures, obligations, three technology ledgers, and three objective ledgers produces nine decision certificates. Each certificate records the selected witness, runner-up, margin, active bottleneck, coordinate provenance, and dependency set.

The three technology rows expose three distinct legal realizations: fused materialization under constrained planar service, split materialization under advanced 2.5D service, and vertical split recomputation after the vertical primitive enters the resource contract. The reference IR therefore constructs the left side of the CRG ladder rather than receiving an anonymous candidate list.

21.2 Head-to-head retention experiment

Experimental construction and interpretation. The experiment uses a declared six-realization family and a registered ledger of post-commit technology and objective shifts. The CRG declared-ledger set is constructed to intersect the argmin sets of that ledger. The comparison therefore demonstrates decision-class-aware retention against winner-only and an initial two-point Pareto projection of equal cardinality; it is not a comparison with every robust or production design-space-exploration policy.

The retention experiment fixes the same six-realization family and compares four reduction maps. Winner-only retention keeps the initial optimum. Initial-PPA Pareto retention keeps the two points nondominated under the registered initial performance, power, and area projection. CRG declared-ledger retention keeps the smallest two-realization family that intersects every argmin set in the declared future price/objective ledger. CRG universal-monotone retention keeps the five realizations whose signatures generate the full finite upper obligation region.

For a retained family $\mathcal{A}_j \subseteq \mathcal{P}_{\text{ref}}$ and post-commit technology/objective point θ , define

$$\text{Regret}_{\theta}(\mathcal{A}_j) = \min_{P \in \mathcal{A}_j} C_{\theta}(P) - \min_{P \in \mathcal{P}_{\text{ref}}} C_{\theta}(P). \quad (21.4)$$

A shifted winner is missed exactly when this quantity is positive.

A named advanced-2.5D shift changes the full-family winner from fused materialization to split materialization. Winner-only and initial-Pareto retention omit the later winner and incur 12.678% relative regret. Both CRG policies retain it and incur zero regret. Across 250 randomized post-commit changes of prices and objective weights, with legality held fixed, the result is:

Retention breadth determines post-change decision coverage

The experiment uses the same six-class registry for all policies across 250 randomized post-commit shifts.

Policy	Retained	Winner-miss rate	Mean regret
Winner-only DSE	1	58.0%	7.018%
Pareto DSE	2	57.6%	6.965%
CRG declared-ledger	2	0.4%	0.021%
CRG universal monotone	5	0.0%	0.000%

Key result: the two-realization CRG declared ledger cuts shifted-winner misses from about 58% to 0.4% while retaining no more designs than the initial Pareto baseline.

Figure 18: Head-to-head retention test. The declared CRG ledger preserves the post-commit decision at the same two-design cardinality as the initial-PPA Pareto reference policy.

Table 13: Retention under 250 post-commit changes. Regret is measured against the full six-realization family.

Policy	Retained	Shifted-winner miss rate	Mean relative regret
Winner-only reference DSE	1	58.0%	7.018%
Initial-PPA Pareto reference DSE	2	57.6%	6.965%
CRG declared ledger	2	0.4%	0.021%
CRG universal monotone	5	0.0%	0.000%

The two two-realization policies expose the content of the theorem. Their cardinality is identical; their decision information is not. The initial-PPA set preserves one starting projection, whereas the declared CRG ledger preserves the argmin structure of the stated future decision family.

A second experiment changes capability rather than price. The vertical class is absent from the original legal family and becomes admissible only after the vertical primitive is introduced. Repricing the old family yields 31.484% relative regret. Reopening the resource contract, generating the vertical realization, and rebuilding the geometry restores the global optimum. The experiment therefore distinguishes movement within a CRG slice from movement to a new slice.

21.3 Carrying one semantic obligation into routed 2.5D and 3D models

The physical study begins from one semantic statement: disjoint ownership of the two stages requires the hidden state to cross their boundary. Two technology contracts serve the same obligation through different embeddings: a side-by-side 2.5D specialist realization and a face-to-face 3D specialist realization. A negotiated A* router uses a 0.5 mm package grid and four routing layers.

For routed net bundle e , the compact timing model is

$$L_e = 10^9 \frac{q_e}{B_e} + L_e^{\text{prop}} + L_e^{\text{proto}}, \quad L_e^{\text{prop}} = \begin{cases} 0.015 \ell_e, & e \text{ planar,} \\ 0.02, & e \text{ vertical,} \end{cases} \quad (21.5)$$

where q_e is bytes per operation, B_e is allocated bytes per second, and ℓ_e is routed planar length in millimeters; the factor 10^9 expresses serialization time in nanoseconds. Link power is the transported bit rate times the declared energy per bit. Signal bumps and port-row count determine required shoreline.

The steady-state thermal field solves the finite-difference conductance system

$$G_{\text{th}}T = P + G_{\text{amb}}T_{\text{amb}}, \quad (21.6)$$

with registered lateral, vertical, and ambient conductances. The PDN field solves the resistive mesh system

$$G_{\text{pdn}}v = s - i, \quad (21.7)$$

with edge supply and a declared vertical-bump delivery drop for upper tiers. For a component of area A , defect density D_0 , and clustering parameter α , the negative-binomial yield coordinate is

$$Y(A; D_0, \alpha) = \left(1 + \frac{D_0 A}{\alpha}\right)^{-\alpha}, \quad (21.8)$$

and system yield multiplies die, interposer, attachment, and vertical-bond terms. The test coordinate is normalized access-lane service with a stack-depth factor. The reliability coordinate combines Arrhenius temperature acceleration

$$A_F(T) = \exp \left[\frac{E_a}{k_B} \left(\frac{1}{T_{\text{ref}}} - \frac{1}{T} \right) \right] \quad (21.9)$$

with routed-interconnect and stack penalties.

Table 14: Deterministic routed comparison for the same semantic ownership contract.

Metric	2.5D side-by-side	3D face-to-face
Package area	1444 mm ²	1122 mm ²
Active silicon	528 mm ²	512 mm ²
Modeled power	278.63 W	274.53 W
Semantic-bridge latency	2002.11 ns	256.62 ns
Critical-link latency	2002.58 ns	2002.13 ns
Maximum temperature	49.91°C	54.04°C
Required shoreline	3.658 mm	1.467 mm
Test-observability score	1.000	0.905
Yield proxy	0.455	0.494

The vertical service reduces bridge latency by 87.2%, shoreline by 59.9%, and package area by 22.3%. HBM service remains near 2002 ns in both realizations, so the critical-link latency remains essentially unchanged. At the same time, vertical integration raises the maximum modeled temperature, concentrates routes, reduces test observability, and changes the reliability-risk coordinate. The physical result is therefore geometric: improving the service of one semantic obligation exposes a different active face of the joint resource system.

21.4 Integrated computation-to-technology statement

The three reference results instantiate the theoretical architecture:

$$\underbrace{\text{IR construction}}_{(F, \rho, \mathcal{T}), \mathcal{P}, \sigma, \mathcal{R}} \longrightarrow \underbrace{\text{safe reduction}}_{\Gamma_{\mathcal{A}}, \mathcal{K}_{\mathcal{A}}} \longrightarrow \underbrace{\text{physical selection}}_{\tau, \phi, \Phi_{\tau}}. \quad (21.10)$$

The graph-to-certificate implementation constructs the legal family and typed signatures; the retention experiment tests decision sufficiency under post-commit change; and the routed study maps one semantic obligation into coupled physical coordinates. They form a complete executable reference path through computation-realization geometry.

Part VI

Synthesis and extension

22 A unified evidence architecture

The theory combines formal mathematics, constructive computation, deterministic measurement, finite-model certification, imported capability anchors, declared coordinates, and physical modeling. Each contributes a different part of the scientific chain.

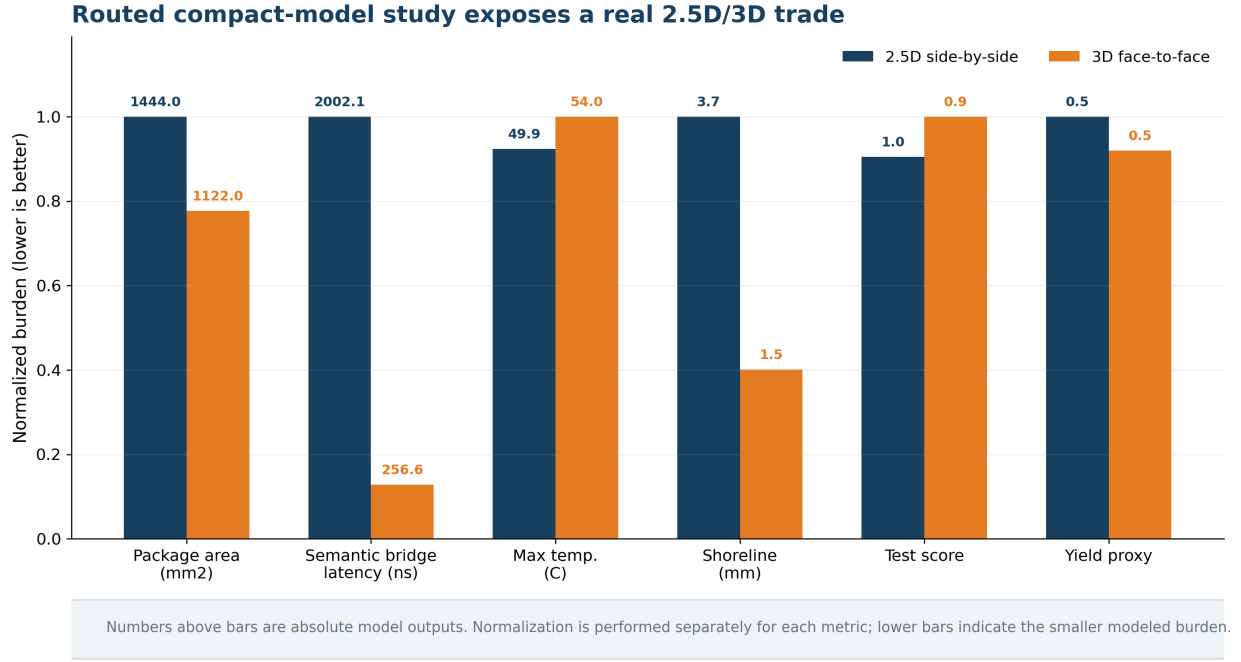


Figure 19: Routed heterogeneous-integration tradeoff. The vertical embedding changes bridge service, shoreline, area, temperature, test, and reliability coordinates while HBM remains the active critical path.

Table 15: Evidence architecture used throughout the manuscript.

Class	Scientific function	Rule of composition
Formal result	Establishes a universal implication in a mathematical contract	Preserve hypotheses, quantifier order, hierarchy, and accounting
Constructive witness	Establishes joint attainability by a legal realization	Carry construction, units, resource ledger, and feasibility conditions
Certified finite-model result	Establishes an exact optimum, tie set, margin, or phase label over a complete finite family	Preserve family, evaluator, objective, and technology domain
Empirical diagnostic	Measures a named realization through a deterministic hierarchy and estimator	Carry dataset identity and measurement contract
Imported evidence	Anchors a workload, device, or capability coordinate	State the source-to-target correspondence
Declared coordinate	Defines a phase or sensitivity axis	Retain units, range, and dependency relation
Reference realization	Demonstrates an executable relation among theory objects	Extend through adapters, transformation grammars, and physical models

The outer-obligation/inner-witness pair is the recurrent proof pattern. An outer theorem identifies what every legal realization in a class provides; an inner construction identifies one jointly attainable point. Equality closes a slice of the geometry. A nonzero bracket identifies the exact information needed to close it. A routed or measured construction returns as a refined inner witness without changing the semantic definition of the computation.

23 The scientific object established here

A Communication Theory of Computation Realization is organized by one ordered architecture:

- (i) terminal semantics determine the distinctions that must become available;

- (ii) the resource contract determines which algorithms, protocols, schedules, ownership structures, hierarchies, and embeddings are legal;
- (iii) legal realizations generate an exact joint signature region;
- (iv) declared observation classes induce canonical decision-sufficient quotients of that region;
- (v) the Safe-Commitment Law characterizes lossless reduction of the realization family;
- (vi) hierarchy projections and scale laws expose multiscale structure;
- (vii) technology acts through capability, embedding, service, and cost; and
- (viii) compiler and physical tools return new constructive witnesses to the same versioned object.

This order replaces “select a graph, measure its communication, and optimize its embedding” with a computation-first theory in which legal alternatives and their joint obligations exist before one architecture or technology becomes definitive.

24 Extension domains

24.1 Stochastic, approximate, and reversible realizations

Randomized and approximate computations replace exact contextual equivalence by distributional or distortion-sensitive relations and add error, confidence, expected traffic, tail latency, and coding coordinates. Reversible realizations add logically irreversible transformations and thermodynamic service to the resource contract. The abstraction principle is unchanged: define the legal realization family, identify the decision class, and derive its sufficient geometry.

24.2 Automatic realization-family construction

Compiler dialects, transformation grammars, program families, e-graphs, architecture generators, and symbolic schedulers can construct compact representations of $\mathcal{P}$. Semantic equivalence and resource legality govern admission; inner witnesses and outer obligations organize the search; and the Safe-Commitment Law governs reduction. The reference IR supplies the machine-readable interface among these elements.

24.3 Physical and manufacturing geometry

Technology contracts can add routed congestion, signal and power integrity, clocking, transient thermal fields, mechanical stress, test access, defect and yield models, reliability, supply risk, and economic coordinates. These coordinates enlarge the technology observation of the same pre-technology realization family. A physical tool can therefore refine an embedding or return a new witness while the semantic and resource contracts remain stable.

24.4 Dynamic lifecycle geometry

Workloads, prices, capabilities, and objectives evolve. CRG separates three operations: reprice an unchanged geometry, recalibrate the embedding model, or reopen the legal family when a new capability changes ρ . This gives a formal lifecycle semantics to design reuse, option value, and technology migration.

25 Implications for computation-to-technology design

CRG changes the unit of design from one selected graph to a versioned family of legal realizations and the decision-sufficient geometry derived from it. A computation contract records behavior, ownership, and correctness. A resource contract records legal storage, recomputation, replication, interaction, scheduling, and hierarchy. Compiler and architecture tools construct witnesses. Semantic and physical converses bound the region. Technology models expose feasible faces, and the design decision records the selected witness, runner-up, margin, active constraints, and dependency relations.

This architecture changes the meaning of iteration. A price change is a new observation of the same geometry. A capability change reopens the affected realization grammar. A routed construction refines a witness. A new converse shrinks an outer region. The design record accumulates rather than being replaced by an unrelated point solution.

26 Conclusion

A computation has many legal physical realizations. They distribute local transformation, retained state, and boundary communication differently across space, time, and hierarchy. Computation-realization geometry makes that family the primary scientific object.

The abstraction ladder

$$(F, \rho, \mathcal{T}) \implies \mathcal{P} \xrightarrow{\sigma} \mathcal{R} \longrightarrow \Gamma \longrightarrow \mathcal{K}$$

separates realization identity, exact joint attainability, monotone-decision information, and additive-price information. The Safe-Commitment Law gives the exact criterion for reducing the realization family without changing the future decisions represented by the retained object.

Canonical semantic states, interactive cut-rank geometry, network carriers, cut-state conservation, observation dualities, scaling laws, technology-phase calculus, analytic component certificates, the Transformer registry, the reference intermediate representation, the retention experiment, and the routed heterogeneous-integration study form one continuous derivation from abstract computation to physical consequence. Together they establish a new level of abstraction for physical computation: the resource-conditioned geometry of legal realizations and the law governing when those realizations may be forgotten.

Preserve the smallest realization geometry sufficient for the future decisions that remain.

Part VII

Appendices

A Harmonized notation and terminology

The notation in Table 16 is reserved globally. Local variables may be used only where their scope is unambiguous and they do not overwrite a reserved meaning.

Symbol	Reserved meaning	Harmonization rule
F	Terminalized computation	Tensor-work counts carry stage subscripts, for example W_{attn} and W_{mlp} , rather than redefining the semantic computation
ρ	Resource contract	Never denotes spectral radius or a mode share
$\mathcal{T}$	Recursive accounting hierarchy	A hierarchy is distinct from a routed carrier network and from elapsed time
P	Legal realization or protocol	The Transformer chiplet count is K_{chip} , not P
$\sigma\mathcal{T}(P)$	Typed signature of one realization	Coordinates retain units and evidence tags; capacity, traffic, service, latency, and energy are not interchangeable
$\mathcal{R}_F^\rho(\mathcal{T})$	Exact attainable signature region	Replaces the separate design-paper notation $\mathcal{Z}$
$\Gamma_F^\rho(\mathcal{T})$	Closed upper obligation region	The interface hyperedge set in scaling arguments is E_∂ , never Γ
$\mathcal{K}_F^\rho(\mathcal{T})$	Closed convex linear-cost body	Used only for the declared additive nonnegative-price observation class
$\mathcal{P}$	Registered realization family	A netlist partition is Π_{part}
w	Nonnegative price vector	A finite-registry witness index is j or ω , not w
τ	Technology contract or technology point	Its declared domain is $\mathcal{D}_{\text{tech}}$; $\Theta(\cdot)$ remains asymptotic notation
$\text{spr}(A)$	Spectral radius of matrix A	Replaces the scaling source's overloaded $\rho(A)$
$\omega_j(g)$	Share of interface mode j at scale g	Replaces overloaded mode weights $\rho_j(g)$
$E_\partial, R_\partial(g)$	Inter-component hyperedges and their profile	Distinct from the CRG obligation region Γ
$p, p_{\text{loc}}(g)$	Asymptotic and local scaling exponents	A separation- tail exponent is ϑ ; a technology polynomial is named q
K_{chip}	Number of compute chiplets in a registered case	Layout factors satisfy $K_{\text{chip}} = CT = C'T'$
H, U, S, X, M, V	Registered Transformer architecture classes	Upright sans-serif labels prevent class names from being read as scalar variables or sets

Table 16: Globally reserved notation.

Term	Canonical meaning	Explicit non-meaning
Realization	One legal algorithm, protocol, circuit, schedule, mapping, allocation, and hierarchy instance under ρ	Not an abstract computation or a technology price
Obligation	A typed quantity every in-contract realization must provide or transport	Not necessarily one physical wire or measured event
Witness	A constructive legal point in the realization set	Not proof that no better unregistered realization exists
Certificate	A proof or checked finite-model consequence with assumptions, units, dependencies, validity domain, and falsifiers	Not a confidence score, vendor guarantee, or physical signoff
Phase cell	A connected sign-invariant region with a constant set-valued feasibility, bottleneck, witness, and winner label	Not a claim that unmodeled technology variation is absent
Rent profile	A registered scale projection $R(g)$ of one realization and accounting procedure	Not the complete CRG and not an intrinsic scalar of a circuit independent of hierarchy
Open artifact	The exact versioned archival payload of source, code, inputs, results, tests, licenses, and provenance identified by its public file index, root manifest, and versioned public deposit	Not blanket permission over other releases, superseded drafts, unreleased materials, or third-party content outside its recorded terms

Table 17: Terminology used across the integrated theory, diagnostics, and case study.

B Contract precedence and quantifier order

Every CRG statement is interpreted in the following order.

- (i) Fix the computation F , terminal ownership, and semantic correctness criterion.
- (ii) Fix the resource contract ρ , including allowed primitives, local work, depth, storage, replication or recomputation, approximation, hierarchy rules, and communication accounting.
- (iii) Choose an admissible hierarchy $\mathcal{T}$, then range over all ρ -legal realizations on that hierarchy.
- (iv) Extract the typed signature of each realization. Only after the exact region is defined may an upper closure, convexification, scale projection, or other observation quotient be taken.
- (v) Apply a technology contract τ as a constraint, embedding model, and cost system. A price-only change does not alter the pre-technology legal set; a capability change that enables a new primitive, schedule, storage mode, or hierarchy changes ρ and therefore selects a different CRG slice.
- (vi) State whether the conclusion is universal over legal realizations, constructive for one witness, exact over a finite registry, diagnostic for one measurement procedure, or proposed future integration.

Changing an earlier item invalidates downstream certificates unless a theorem explicitly proves invariance or monotonicity under that change. In particular, an observed phase winner cannot be transferred to a new execution policy or package accounting rule merely because the workload name is unchanged.

C Essential reusable contract clauses

The following clauses prevent the most common overextensions of the integrated results.

C.1 Interaction, traffic, and time

Directional contextual summaries are exact constructive states in the declared noninteractive model. Arbitrary interaction can change attainable nonlinear signatures; the finite-field linear fragment is separately exact by the cross-cut rank theorem. Static transcript capacity is not instantaneous bandwidth or latency. A temporal conclusion additionally requires a service window, retained cut state, event direction, and accounting rule. Joint coding, compression, variable length, retransmission, and multicast are free only when the contract expressly supplies their decoder and charging rules.

C.2 Layout conversion

The aligned-layout assignment law assumes disjoint exhaustive equal-capacity source and target partitions, uniform represented-byte density, one initial and final owner per element, and free bijective chiplet relabeling. It excludes replication, recomputation, coding, compression, cached duplicates, unequal-capacity placement, congestion, and route-dependent cost. The nested closed form requires $C \mid C'$ or $C' \mid C$; the fact that both factors divide K_{chip} is not sufficient. The general result is the maximum-weight assignment, and its minimum is represented-byte movement rather than routed traffic, latency, or energy.

C.3 Scaling and boundary modes

A positive mixing cocycle supplies a sufficient projective mechanism. In that local matrix context its factors are A_k , its period product is M , and its spectral radius is $\text{spr}(M)$, leaving ρ reserved for the resource contract. Mode-mixture weights are $\omega_j(g)$. Inter-component boundary edges are E_∂ and contribute $R_\partial(g)$; they are not the upper obligation region Γ . One-phase dyadic samples cannot establish a nontrivial log-periodic phase function. Failure of a necessary observable rejects a certificate, whereas agreement does not prove unique mechanism or basin membership.

C.4 Finite-registry design decisions

A phase label is set-valued and retains infeasible cells, ties, active bottlenecks, witness exchanges, and architecture exchanges. A retained family is lossless for a technology set precisely when it intersects every attained argmin set. Strict positive regret for every singleton additionally requires the corresponding finite attained ledger. No sampled pairwise reversal alone rules out an unregistered robust compromise.

C.5 Transformer certificate

The primary result concerns batch-one prefill under the named GQA tensor-work, pair-execution, peak-basis, package-sharing, and architecture-registry contracts. Auxiliary kernels and physical coordinates listed as exclusions remain outside the objective. The split bridge is semantic only under disjoint single ownership and the declared steady-state exchange. The universal and static classes use different state aggregators because one reuses state across disjoint times and the other provisions dedicated state simultaneously. Imported FlatAttention coordinates are sensitivity variables, not GQA calibration facts.

D Worked derivations of the CRG abstraction ladder

This appendix derives the abstraction ladder in elementary finite examples. The examples are intentionally small: they show exactly what each object retains, why the objects cannot be interchanged, and how the same formalism connects semantic obligation to a technology-conditioned decision.

D.1 Broadcast and parity: semantic state versus physical incidence

Let a tree carry one binary source x at vertex s . In the broadcast task, every terminal in a set D must output x . Delete an edge e and orient it from the component containing s toward the opposite component. If the opposite component contains at least one demanded output, then two source-side assignments $x = 0$ and $x = 1$ induce different required outputs there. The contextual quotient therefore has two classes and

$$\kappa_e = \log_2 2 = 1.$$

The reverse quotient has one class because the source-side outputs do not depend on opposite-side inputs. Hence one semantic bit is required in the forward direction on every edge separating the source from at least one consumer.

The same bit can cross many hierarchy interfaces. If the source is at the root of a balanced tree and all leaves consume the bit, the sum of edge incidences grows with the distribution tree although the semantic source contains only one bit. Information content and multiscale physical incidence are therefore different coordinates.

Now place one source bit x_v at every vertex and require every vertex to output the global parity

$$y = \bigoplus_{v \in V} x_v.$$

Across any nontrivial cut $A \mid B$, the A side contributes one local parity $p_A = \bigoplus_{v \in A} x_v$ and the B side contributes p_B . To produce $y = p_A \oplus p_B$ on both sides, one bit must cross in each direction. The canonical signature on every oriented edge is therefore $(1, 1)$ in the two directions. Broadcast and parity can both involve one-bit summaries, yet their directionality and hierarchy incidence differ. CRG retains those differences rather than compressing them into one undirected communication number.

D.2 INDEX: interaction changes the attainable geometry

Let Alice hold $x \in \{0, 1\}^n$, let Bob hold $i \in [n]$, and require Bob to output x_i . Under a one-way Alice-to-Bob contract, Alice's message must distinguish every pair $x \neq x'$. Indeed, choose an index i at which they differ; Bob must then produce different outputs. Thus every valid one-way message alphabet has at least 2^n states, and the one-way signature is

$$r_{\text{one}} = (n, 0)$$

in coordinates $(A \rightarrow B, B \rightarrow A)$.

Under a two-round interactive contract, Bob sends i using $\lceil \log_2 n \rceil$ bits and Alice returns one bit x_i . This gives the constructive signature

$$r_{\text{int}} = (1, \lceil \log_2 n \rceil).$$

The exact region contains at least these two incomparable points:

$$\mathcal{R} \supseteq \{r_{\text{one}}, r_{\text{int}}\}.$$

For a nonnegative price $w = (w_{AB}, w_{BA})$, their costs are

$$C_w(r_{\text{one}}) = nw_{AB}, \quad C_w(r_{\text{int}}) = w_{AB} + \lceil \log_2 n \rceil w_{BA}.$$

The interactive realization wins precisely when

$$(n-1)w_{AB} > \lceil \log_2 n \rceil w_{BA},$$

and the one-way realization wins in the reverse regime. The computation has no price-independent preferred protocol. Interaction is a legal-realization choice, and technology or policy prices expose different points of the same realization geometry.

D.3 Exact points, upper obligations, and price-visible bodies

Consider a finite realization family with two exact signatures

$$r = (0, 2), \quad s = (2, 0).$$

The exact attainable region is

$$\mathcal{R} = \{r, s\}.$$

Its upper obligation region is

$$\Gamma = \text{up } \mathcal{R} = (r + \mathbb{R}_+^2) \cup (s + \mathbb{R}_+^2),$$

and its convex price-visible body is

$$\mathcal{K} = \text{cl conv } \Gamma = \text{conv}\{r, s\} + \mathbb{R}_+^2.$$

The midpoint $(1, 1)$ belongs to $\mathcal{K}$ but not to $\mathcal{R}$ or Γ . Therefore the monotone objective

$$\Phi(x) = \max\{x_1, x_2\}$$

has

$$\inf_{x \in \mathcal{R}} \Phi(x) = 2, \quad \inf_{x \in \mathcal{K}} \Phi(x) = 1.$$

Convexification is exact for additive prices, because every linear functional attains its infimum on an extreme point of the segment, but it is not exact for all monotone objectives. This is the elemental reason that $\mathcal{R}$, Γ , and $\mathcal{K}$ are distinct scientific objects.

The example also shows why coordinatewise minima can be fictitious. The minimum of each coordinate is zero, but no legal realization attains $(0, 0)$. A report that lists only marginal minima silently combines the first coordinate of one design with the second coordinate of another.

D.4 Safe commitment and explicit regret

Retain only the first realization, $\mathcal{A} = \{r\}$. For the price $w = (1, 3)$,

$$C_w(r) = 6, \quad C_w(s) = 2.$$

The full family selects s , while the retained family is forced to select r . Commitment regret is

$$\text{Regret}_w(\mathcal{A}) = \inf_{a \in \mathcal{A}} \langle w, a \rangle - \inf_{x \in \mathcal{R}} \langle w, x \rangle = 6 - 2 = 4.$$

Equivalently, $\mathcal{K}_{\mathcal{A}} \neq \mathcal{K}$. The linear-cost part of the Safe-Commitment Law therefore predicts the existence of a separating price, and this price realizes the prediction explicitly.

If the future decision class consists only of the single price $w = (3, 1)$, then retaining r is sufficient because r is the unique minimizer for that one decision. If every nonnegative price remains possible, retaining only r is insufficient. Safe retention is thus indexed by the future decision class, not by an unconditional requirement to preserve every realization.

D.5 From a compiler graph to CRG objects

The reference intermediate representation instantiates the formal chain as follows. A compiler graph supplies operators, tensor dependencies, shapes, and terminal ownership. A computation contract fixes the required outputs and semantic equivalence. A resource contract declares legal transformations such as materialization, recomputation, fusion, split ownership, replication, and vertical placement. Applying those transformations generates a finite registered family

$$\mathcal{P}_{\text{ref}} = \{P_1, \dots, P_6\}.$$

For each P_j , the extractor computes a typed signature

$$\sigma(P_j) = (W_j, M_j, Q_j^{\text{HBM}}, Q_j^{\text{pkg}}, D_j, E_j, T_j, \dots).$$

The six vectors form the exact registered image $\mathcal{R}_{\text{ref}}$. Dominance closure produces the monotone region used for resource-feasibility queries; convexification produces the body used for additive prices. Scoped outer bounds enter as inequalities satisfied by every legal member of a declared class. The decision certificate then records

$$P^* \in \underset{P \in \mathcal{P}_{\text{ref}}}{\operatorname{argmin}} C_{\tau}(P),$$

together with the runner-up, margin, active constraints, witness data, and reopen dependencies.

This mapping explains the role of the reference implementation. It is not a separate empirical object attached to the theory; it is an executable construction of $(F, \rho, \mathcal{T})$, $\mathcal{P}$, σ , $\mathcal{R}$, scoped outer obligations, and a technology-conditioned selection from the same formal chain.

D.6 Repricing versus reopening

Let $\mathcal{P}$ and its signatures be fixed. Replacing one nonnegative price vector w by another w' changes

$$\underset{P \in \mathcal{P}}{\operatorname{argmin}} \langle w, \sigma(P) \rangle$$

without changing $\mathcal{P}$, $\mathcal{R}$, Γ , or $\mathcal{K}$. The design family can be repriced.

Now suppose a new vertical primitive becomes legal. The resource contract changes from ρ to ρ' with

$$\mathcal{P}_F^{\rho}(\mathcal{T}) \subsetneq \mathcal{P}_F^{\rho'}(\mathcal{T}).$$

A new signature may lie outside the old geometry and dominate every old candidate for the updated technology. No price sweep over the old body can recover it. The correct operation is to reopen the legal family, construct the new geometry, and then select. This is the formal distinction between a technology price update and a technology capability update.

D.7 Why monotone observations reconstruct the upper region

Let $\Gamma \subseteq \mathbb{R}_+^d$ be closed and upper closed. For every test point $a \in \mathbb{R}_+^d$, define

$$\Phi_a(x) = \|(x - a)_+\|_{\infty}, \quad (x - a)_+ = (\max\{x_i - a_i, 0\})_{i=1}^d.$$

This function is continuous and coordinatewise nondecreasing. If $a \in \Gamma$, then $\Phi_a(a) = 0$, so $\inf_{x \in \Gamma} \Phi_a(x) = 0$. Conversely, if the infimum is zero, choose $x_n \in \Gamma$ with $\Phi_a(x_n) \rightarrow 0$ and define $y_n = x_n \vee a$ coordinatewise. Upper closure gives $y_n \in \Gamma$ and $y_n \rightarrow a$; closedness gives $a \in \Gamma$. Hence

$$a \in \Gamma \iff \inf_{x \in \Gamma} \Phi_a(x) = 0.$$

The complete response to continuous monotone objectives therefore determines membership of every point and reconstructs the entire upper region. This is the key step behind the monotone part of the Safe-Commitment Law.

For a retained family $\mathcal{A}$, suppose $\Gamma_{\mathcal{A}} \subsetneq \Gamma$. Choose $a \in \Gamma \setminus \Gamma_{\mathcal{A}}$. Then the full family has value zero for Φ_a , whereas the retained family has strictly positive value. The separating objective is constructed directly from the missing point; positive commitment regret is not an appeal to an unspecified future cost.

D.8 Why additive prices reconstruct the convex upper body

Let $\mathcal{K} \subseteq \mathbb{R}_+^d$ be nonempty, closed, convex, and upper closed. Its lower support response is

$$h_{\mathcal{K}}(w) = \inf_{x \in \mathcal{K}} \langle w, x \rangle, \quad w \in \mathbb{R}_+^d.$$

Every $x \in \mathcal{K}$ satisfies $\langle w, x \rangle \geq h_{\mathcal{K}}(w)$ for all nonnegative w . If $x \notin \mathcal{K}$, strong separation supplies a vector a and scalar α with

$$\langle a, x \rangle < \alpha \leq \langle a, y \rangle \quad \text{for all } y \in \mathcal{K}.$$

Upper closure forces $a \geq 0$: if $a_i < 0$, then $y + te_i \in \mathcal{K}$ for every $t \geq 0$ and $\langle a, y + te_i \rangle \rightarrow -\infty$, contradicting separation. Therefore a is an admissible nonnegative price, and

$$\langle a, x \rangle < h_{\mathcal{K}}(a).$$

It follows that

$$\mathcal{K} = \{x \in \mathbb{R}_+^d : \langle w, x \rangle \geq h_{\mathcal{K}}(w) \text{ for every } w \geq 0\}.$$

Thus the complete additive-price response determines $\mathcal{K}$ exactly. The convexification step is justified precisely because linear prices cannot distinguish a raw set from its convex hull, while nonlinear monotone objectives can.

D.9 A two-architecture phase wall

Consider two legal realizations with technology-dependent costs

$$C_A(\theta) = \max\{a_1(\theta), a_2(\theta)\}, \quad C_B(\theta) = \max\{b_1(\theta), b_2(\theta)\},$$

where all branches are rational functions on a semialgebraic domain Θ . The winner can change only when one of the following equalities is met:

$$a_1 = a_2, \quad b_1 = b_2, \quad C_A = C_B,$$

or when a feasibility inequality becomes tight. Clearing denominators on regions of fixed denominator sign converts these conditions into polynomial sign tests. A sign-invariant cell decomposition therefore partitions Θ into finitely many regions on which feasibility, active bottlenecks, and the complete winner set are constant.

At a point θ_0 away from every wall, let $p_1, \dots, p_m$ be the cleared polynomials used by the decision and let L_i bound the local Lipschitz variation of p_i . Then

$$r < \min_i \frac{|p_i(\theta_0)|}{L_i}$$

preserves every sign and hence the complete phase label. The radius is a certificate for the full decision record, not merely the distance to the nearest pairwise architecture tie.

D.10 Deriving the architecture–technology scaling law

Let one realization have demand

$$R(g) = g^p \ell_R(g)$$

and one physical technology have boundary service

$$B_{\tau}(g) = g^q \ell_B(g),$$

with ℓ_R and ℓ_B slowly varying. The dimensionless strain is

$$\chi_{\tau}(g) = \frac{R(g)}{B_{\tau}(g)} = g^{p-q} \frac{\ell_R(g)}{\ell_B(g)}.$$

The quotient of positive slowly varying functions is slowly varying, so $\chi_{\tau} \in \text{RV}_{p-q}$. The sign of $p - q$ controls the asymptotic direction:

$$p < q \Rightarrow \chi_{\tau}(\lambda g) / \chi_{\tau}(g) < 1 \text{ at large scale,}$$

$$p > q \Rightarrow \chi_{\tau}(\lambda g) / \chi_{\tau}(g) > 1,$$

and $p = q$ leaves the decision to constants and slowly varying factors.

For an ideal homogeneous d -dimensional body, volume grows as g and boundary measure as $g^{(d-1)/d}$, hence $q = (d-1)/d$. Two-dimensional and three-dimensional homogeneous service therefore have marginal indices $1/2$ and $2/3$. This derivation is a scale-matching law: the complete physical technology model supplies routing, timing, power-delivery, thermal, test, yield, and reliability constraints in addition to the boundary-service index.

D.11 Transformer bridge arithmetic from ownership semantics

For context length N , hidden dimension d , and b_{elem} bytes per element, one hidden-state tensor occupies

$$A(N) = Ndb_{\text{elem}}.$$

If attention and feed-forward stages have disjoint single-owner regions, the state must move from attention to feed-forward and the next state must return to the attention owner in steady operation. The two-way semantic bridge is therefore

$$Q_B(N) = 2Ndb_{\text{elem}}.$$

For $N = 16,384$, $d = 4096$, and BF16/FP16 $b_{\text{elem}} = 2$ bytes,

$$A(N) = 16,384 \times 4096 \times 2 = 134,217,728 \text{ bytes} = 128 \text{ MiB},$$

$$Q_B(N) = 256 \text{ MiB}.$$

At a declared payload of 512 GB/s, the serialization time of the bridge is

$$\frac{256 \text{ MiB}}{512 \text{ GB/s}} = 0.524288 \text{ ms}$$

under the decimal-rate/binary-size convention used by the registered model. The calculation begins from ownership semantics; physical placement determines how the obligation is served and priced.

The state-feasibility distinction follows from two different resource aggregators. If dedicated attention and feed-forward engines require 28 MiB and 24 MiB simultaneously, static provisioning requires 52 MiB. If one universal engine executes the stages in disjoint time intervals and reuses the same state pool, the required capacity is $\max\{28, 24\} = 28$ MiB. The designs are not two estimates of one architecture; they are two legal realizations with different time and efficiency coordinates.

D.12 Deriving retention regret in the reference experiment

Let P_0^* be the winner under the initial technology and let P_1^* be the winner after the advanced-2.5D price update. A winner-only policy retains $\mathcal{A} = \{P_0^*\}$. If the updated costs are $C_1(P_0^*)$ and $C_1(P_1^*)$, relative regret is

$$\text{RelRegret}_1(\mathcal{A}) = \frac{C_1(P_0^*) - C_1(P_1^*)}{C_1(P_1^*)}.$$

The registered values give 12.678%. The CRG declared ledger contains P_1^* , so its updated regret is zero. Across the randomized ledger, the miss rate is the empirical frequency of

$$\mathcal{A} \cap \underset{P \in \mathcal{P}_{\text{ref}}}{\text{argmin}} C_\theta(P) = \emptyset.$$

This is the finite-domain condition of argmin-hitting retention evaluated over sampled future technology and objective points.

When a vertical capability adds a new legal realization $P_V \notin \mathcal{P}_{\text{old}}$, every retained subset of the old family misses P_V before reopening. The resulting 31.484% regret is not a failure of price coverage inside the old geometry; it is evidence that the legal family has changed. Reopening constructs $\mathcal{P}_{\text{new}} = \mathcal{P}_{\text{old}} \cup \{P_V\}$ and restores the correct technology-conditioned optimum.

E Derivation map from theorem to engineering artifact

Table 18: How the principal formal objects appear in an engineering record.

Formal object	Mathematical content	Engineering artifact
F	Terminalized computation and correctness relation	Workload/operator graph, output ownership, exact or approximate semantics

Formal object	Mathematical content	Engineering artifact
ρ	Legal primitives, resource bounds, accounting, hierarchy rules	Transformation rule set, memory/recompute policy, replication rules, precision and scheduling contract
$\mathcal{P}$	Legal realizations modulo internal-state renaming	Registered algorithms, dataflows, schedules, mappings, partitions, layouts, and allocations
$\sigma(P)$	Typed joint obligation vector	Work, state, traffic, events, depth, latency, activity, energy and physical-interface fields with units
$\mathcal{R}$	Exact jointly attainable signatures	Candidate database retaining discrete attainable combinations and realization provenance
Γ	Closed upper obligation region	Complete representation for monotone resource feasibility and monotone cost queries
$\mathcal{K}$	Convex upper price-visible body	Complete representation for additive nonnegative price sweeps and admitted mixtures
Outer obligation	Converse satisfied by every realization in a class	Certified lower bound or infeasibility certificate
Inner witness	One explicit legal construction	Compiler schedule, RTL/netlist, routed package, measured kernel, or other constructive evidence
Safe-Commitment Law	Equality of the required decision-sufficient object before and after reduction	Pruning certificate and retained-alternative ledger
Technology point τ	Capability, embedding, service, and cost model	PDK/package/library coordinates, calibrated models, and objective weights
Decision certificate	Argmin, margin, dependencies, and active constraints	Architecture-review record with repricing and reopen triggers

References

- [1] H. T. Kung, “The structure of parallel algorithms,” *Advances in Computers*, vol. 19, pp. 65–112, 1980.
- [2] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.
- [3] M. Braverman, “Interactive information complexity,” *SIAM Journal on Computing*, vol. 44, no. 6, pp. 1698–1739, 2015.
- [4] B. Barak, M. Braverman, X. Chen, and A. Rao, “How to compress interactive communication,” *SIAM Journal on Computing*, vol. 42, no. 3, pp. 1327–1363, 2013.
- [5] J.-W. Hong and H. T. Kung, “I/O complexity: The red-blue pebble game,” in *Proceedings of the 13th ACM Symposium on Theory of Computing*, pp. 326–333, 1981.
- [6] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz, “Minimizing communication in numerical linear algebra,” *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 3, pp. 866–901, 2011.
- [7] M. Frigo, C. E. Leiserson, H. Prokop, and S. Ramachandran, “Cache-oblivious algorithms,” in *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pp. 285–297, 1999.
- [8] D. Irony, S. Toledo, and A. Tiskin, “Communication lower bounds for distributed-memory matrix multiplication,” *Journal of Parallel and Distributed Computing*, vol. 64, no. 9, pp. 1017–1026, 2004.
- [9] E. Solomonik and J. Demmel, “Communication-optimal parallel 2.5D matrix multiplication and LU factorization algorithms,” in *Euro-Par 2011*, pp. 90–109, 2011.
- [10] UCIE Consortium, “UCIE 3.0 Specification,” 2025. <https://www.uciexpress.org/specifications>.
- [11] UCIE Consortium, “UCIE 2.0 White Paper,” 2024, accessed through the official specification portal. <https://www.uciexpress.org/specifications>.

- [12] AMD, “4th Gen AMD EPYC Processor Architecture,” white paper, 2023. <https://www.amd.com/content/dam/amd/en/documents/products/epyc/4th-gen-epyc-processor-architecture-white-paper.pdf>.
- [13] R. A. van de Geijn and J. Watts, “SUMMA: Scalable universal matrix multiplication algorithm,” *Concurrency: Practice and Experience*, vol. 9, no. 4, pp. 255–274, 1997.
- [14] R. W. Hockney, “The communication challenge for MPP: Intel Paragon and Meiko CS-2,” *Parallel Computing*, vol. 20, no. 3, pp. 389–398, 1994.
- [15] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, “FlashAttention: Fast and memory-efficient exact attention with IO-awareness,” in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 16344–16359. Available: <https://arxiv.org/abs/2205.14135>.
- [16] NVIDIA, “NVIDIA A100 Tensor Core GPU,” product datasheet and architecture white paper, 2020–2021. <https://www.nvidia.com/en-us/data-center/a100/>.
- [17] L. Zheng et al., “Alpa: Automating inter- and intra-operator parallelism for distributed deep learning,” in *OSDI*, pp. 559–578, 2022.
- [18] D. Lepikhin et al., “GSPMD: General and scalable parallelization for ML computation graphs,” arXiv:2105.04663, 2021. <https://arxiv.org/abs/2105.04663>.
- [19] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.
- [20] A. M. Turing, “On computable numbers, with an application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, series 2, vol. 42, pp. 230–265, 1936.
- [21] A. C.-C. Yao, “Some complexity questions related to distributive computing,” in *Proceedings of the 11th ACM Symposium on Theory of Computing*, pp. 209–213, 1979.
- [22] E. Kushilevitz and N. Nisan, *Communication Complexity*. Cambridge University Press, 1997.
- [23] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [24] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, “Network coding for computing: Cut-set bounds,” *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1015–1030, 2011.
- [25] H. Kowshik and P. R. Kumar, “Optimal function computation in directed and undirected graphs,” *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3407–3418, 2012.
- [26] C. D. Thompson, “Area-time complexity for VLSI,” in *Proceedings of the 11th ACM Symposium on Theory of Computing*, pp. 81–88, 1979.
- [27] L. G. Valiant, “A bridging model for parallel computation,” *Communications of the ACM*, vol. 33, no. 8, pp. 103–111, 1990.
- [28] D. Culler et al., “LogP: Towards a realistic model of parallel computation,” in *Proceedings of the 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp. 1–12, 1993.
- [29] B. S. Landman and R. L. Russo, “On a pin versus block relationship for partitions of logic graphs,” *IEEE Transactions on Computers*, vol. C-20, no. 12, pp. 1469–1479, 1971, doi:10.1109/T-C.1971.223159.
- [30] P. Christie and D. Stroobandt, “The interpretation and application of Rent’s rule,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 6, pp. 639–648, 2000.
- [31] K. Keutzer, S. Malik, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, “System-level design: Orthogonalization of concerns and platform-based design,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1523–1543, 2000, doi:10.1109/43.898830.
- [32] A. Sangiovanni-Vincentelli, L. P. Carloni, F. De Bernardinis, and M. Sgroi, “Benefits and challenges for platform-based design,” in *Proceedings of the 41st Design Automation Conference*, pp. 409–414, 2004.

- [33] L. de Alfaro and T. A. Henzinger, “Interface automata,” in *Proceedings of the 8th European Software Engineering Conference*, pp. 109–120, 2001, doi:10.1145/503209.503226.
- [34] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. A. Henzinger, and K. G. Larsen, *Contracts for System Design*. Foundations and Trends in Electronic Design Automation, vol. 12, no. 2–3, pp. 124–400, 2018, doi:10.1561/10000000053.
- [35] A. Löhne, *Vector Optimization with Infimum and Supremum*. Springer, 2011.
- [36] A. Löhne, B. Rudloff, and F. Ulus, “Primal and dual approximation algorithms for convex vector optimization problems,” *Journal of Global Optimization*, vol. 60, no. 4, pp. 713–736, 2014, doi:10.1007/s10898-013-0136-0.
- [37] A. Parashar et al., “Timeloop: A systematic approach to DNN accelerator evaluation,” in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2019. https://research.nvidia.com/publication/2019-03_timeloop-systematic-approach-dnn-accelerator-evaluation.
- [38] H. Kwon, P. Chatarasi, M. Pellauer, A. Parashar, V. Sarkar, and T. Krishna, “Understanding reuse, performance, and hardware cost of DNN dataflows: A data-centric approach using MAESTRO,” in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 754–768, 2019. <https://arxiv.org/abs/1805.02566>.
- [39] L. Mei, P. Houshmand, V. Jain, S. Giraldo, and M. Verhelst, “ZigZag: Enlarging joint architecture–mapping design space exploration for DNN accelerators,” *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1160–1174, 2021. <https://arxiv.org/abs/2007.11360>.
- [40] Q. Huang et al., “CoSA: Scheduling by constrained optimization for spatial accelerators,” in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2021. <https://arxiv.org/abs/2105.01898>.
- [41] J. Cai et al., “Gemini: Mapping and architecture co-exploration for large-scale DNN chiplet accelerators,” in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024. <https://arxiv.org/abs/2312.16436>.
- [42] A. B. Kahng, “OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain,” in *Proceedings of the ACM International Symposium on Physical Design*, 2019.
- [43] Y. N. Wu, J. S. Emer, and V. Sze, “Accelerger: An architecture-level energy estimation methodology for accelerator designs,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019. <https://accelergy.mit.edu/>.
- [44] N. Dehghani and G. Caterina, “Physical Computing: A Category Theoretic Perspective on Physical Computation and System Compositionality,” *Journal of Physics: Complexity*, 2024, doi:10.1088/2632-072X/ad6051.
- [45] R. Landauer, “Irreversibility and heat generation in the computing process,” *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183–191, 1961.
- [46] J. Bochnak, M. Coste, and M.-F. Roy, *Real Algebraic Geometry*. Springer, 1998.
- [47] The OpenROAD Project, “OpenROAD documentation,” accessed July 2026. <https://openroad.readthedocs.io/>.
- [48] S. Kang et al., “ArtNet: Hierarchical clustering-based artificial netlist generator for ML and DTCO applications,” preprint arXiv:2510.13582, 2025.
- [49] L. Amarù, P.-E. Gaillardon, and G. De Micheli, “The EPFL combinational benchmark suite,” in *Proceedings of IWLS*, 2015.
- [50] G. Birkhoff, “Extensions of Jentzsch’s theorem,” *Transactions of the American Mathematical Society*, vol. 85, pp. 219–227, 1957.
- [51] P. J. Bushell, “Hilbert’s metric and positive contraction mappings in a Banach space,” *Archive for Rational Mechanics and Analysis*, vol. 52, pp. 330–338, 1973.
- [52] G. Karypis and V. Kumar, “Multilevel k -way hypergraph partitioning,” in *Proceedings of the 36th ACM/IEEE Design Automation Conference*, pp. 343–348, 1999, doi:10.1145/309847.309954.

- [53] N. Selvakumaran and G. Karypis, “Multi-objective hypergraph partitioning algorithms for cut and maximum subdomain degree minimization,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 726–733, 2003, doi:10.1109/ICCAD.2003.159757.
- [54] N. H. Bingham, C. M. Goldie, and J. L. Teugels, *Regular Variation*. Cambridge University Press, 1987, doi:10.1017/CBO9780511721434.
- [55] D. Sornette, “Discrete-scale invariance and complex dimensions,” *Physics Reports*, vol. 297, no. 5, pp. 239–270, 1998, doi:10.1016/S0370-1573(97)00076-8.
- [56] C. Bornstein, A. Litman, B. Maggs, R. Sitaraman, and T. Yatzkar, “On the bisection width and expansion of butterfly networks,” in *Proceedings of IPPS/SPDP*, 1998.
- [57] A. Grattafiori *et al.*, “The Llama 3 herd of models,” arXiv:2407.21783, 2024. Available: <https://arxiv.org/abs/2407.21783>.
- [58] Meta Llama, “Llama 3 reference implementation and model parameters,” 2024. <https://github.com/meta-llama/llama3>.
- [59] NVIDIA, “GB200 NVL72,” product information, accessed July 2026. <https://www.nvidia.com/en-us/data-center/gb200-nvl72/>.
- [60] Micron Technology, “Micron HBM3E,” product information, accessed July 2026. <https://www.micron.com/products/memory/hbm/hbm3e>.
- [61] C. Zhang *et al.*, “FlatAttention: Dataflow and fabric collectives co-optimization for efficient multi-head attention on tile-based many-PE accelerators,” arXiv:2505.18824, 2025. <https://arxiv.org/abs/2505.18824>.
- [62] R. Ahlswede and N. Cai, “On communication complexity of vector-valued functions,” *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 2062–2067, 1994.
- [63] R. Dougherty, C. Freiling, and K. Zeger, “Networks, matroids, and non-Shannon information inequalities,” *IEEE Transactions on Information Theory*, vol. 53, no. 6, pp. 1949–1969, 2007.
- [64] C. T. Li, “Undecidability of network coding, conditional information inequalities, and conditional independence implication,” *IEEE Transactions on Information Theory*, vol. 69, no. 6, pp. 3493–3510, 2023.
- [65] R. Gandy, “Church’s thesis and principles for mechanisms,” in *The Kleene Symposium*, J. Barwise, H. J. Keisler, and K. Kunen, Eds. North-Holland, 1980, pp. 123–148, doi:10.1016/S0049-237X(08)71257-6.
- [66] D. Deutsch, “Quantum theory, the Church–Turing principle and the universal quantum computer,” *Proceedings of the Royal Society of London A*, vol. 400, no. 1818, pp. 97–117, 1985.
- [67] R. E. Kalman, “Mathematical description of linear dynamical systems,” *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 1, no. 2, pp. 152–192, 1963.
- [68] B. L. Ho and R. E. Kalman, “Effective construction of linear state-variable models from input/output functions,” *Regelungstechnik*, vol. 14, pp. 545–548, 1966.
- [69] A. Nerode, “Linear automaton transformations,” *Proceedings of the American Mathematical Society*, vol. 9, no. 4, pp. 541–544, 1958.
- [70] G. Kahn, “The semantics of a simple language for parallel programming,” in *Information Processing 74*, J. L. Rosenfeld, Ed. North-Holland, 1974, pp. 471–475.
- [71] C. Hewitt, P. Bishop, and R. Steiger, “A universal modular ACTOR formalism for artificial intelligence,” in *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, 1973, pp. 235–245.
- [72] E. A. Lee and D. G. Messerschmitt, “Synchronous data flow,” *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1235–1245, 1987.
- [73] D. Blackwell, “Equivalent comparisons of experiments,” *The Annals of Mathematical Statistics*, vol. 24, no. 2, pp. 265–272, 1953.

- [74] P. Cousot and R. Cousot, “Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints,” in *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 1977, pp. 238–252.
- [75] U. Bondhugula, A. Hartono, J. Ramanujam, and P. Sadayappan, “A practical automatic polyhedral parallelizer and locality optimizer,” in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2008, pp. 101–113.
- [76] J. Ragan-Kelley et al., “Halide: A language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines,” in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2013, pp. 519–530.
- [77] T. Chen et al., “TVM: An automated end-to-end optimizing compiler for deep learning,” in *13th USENIX Symposium on Operating Systems Design and Implementation*, 2018, pp. 578–594.
- [78] C. Lattner et al., “MLIR: Scaling compiler infrastructure for domain specific computation,” in *IEEE/ACM International Symposium on Code Generation and Optimization*, 2021, pp. 2–14.
- [79] R. Tate, M. Stepp, Z. Tatlock, and S. Lerner, “Equality saturation: A new approach to optimization,” in *Proceedings of the 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2009, pp. 264–276.
- [80] M. Willsey et al., “egg: Fast and extensible equality saturation,” *Proceedings of the ACM on Programming Languages*, vol. 5, no. POPL, Art. 23, 2021.
- [81] S. Oum and P. Seymour, “Approximating clique-width and branch-width,” *Journal of Combinatorial Theory, Series B*, vol. 96, no. 4, pp. 514–528, 2006.
- [82] W. E. Donath, “Placement and average interconnection lengths of computer logic,” *IEEE Transactions on Circuits and Systems*, vol. 26, no. 4, pp. 272–277, 1979.
- [83] L. Hagen, A. B. Kahng, F. J. Kurdahi, and C. Ramachandran, “On the intrinsic Rent parameter and spectral-based partitioning methodologies,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 1, pp. 27–37, 1994.
- [84] H. Van Marck, D. Stroobandt, and J. Van Campenhout, “Towards an extension of Rent’s rule for describing local variations in interconnection complexity,” in *Proceedings of the International Conference on Young Computer Scientists*, 1995, pp. 136–141.
- [85] P. Zarkesh-Ha, J. A. Davis, and J. D. Meindl, “On a pin versus gate relationship for heterogeneous systems: Heterogeneous Rent’s rule,” in *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1998, pp. 93–96.
- [86] D. Stroobandt, *A Priori Wire Length Estimates for Digital Design*. Kluwer Academic Publishers, 2001.
- [87] C. Mead and L. Conway, “VLSI systems design: A new dimension of computer education,” *Proceedings of the IEEE*, vol. 67, no. 9, pp. 1225–1234, 1979.
- [88] C. Mead and L. Conway, *Introduction to VLSI Systems*. Addison-Wesley, 1980.
- [89] H. P. Benson, “An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem,” *Journal of Global Optimization*, vol. 13, pp. 1–24, 1998.
- [90] C. H. Papadimitriou and M. Yannakakis, “On the approximability of trade-offs and optimal access of Web sources,” in *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, 2000, pp. 86–92.
- [91] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton University Press, 2009.
- [92] H. Qu et al., “MLDSE: Scaling design space exploration infrastructure for multi-level hardware,” arXiv:2503.21297, 2025.
- [93] H. Jin et al., “Mozart: A chiplet ecosystem–accelerator codesign framework for composable bespoke application specific integrated circuits,” arXiv:2510.08873, 2025.

- [94] Y. Shi et al., “Open3DBench: Open-source benchmark for 3D-IC backend implementation and PPA evaluation,” arXiv:2503.12946, 2025, revised 2026.
- [95] C. H. Bennett, “The thermodynamics of computation—a review,” *International Journal of Theoretical Physics*, vol. 21, pp. 905–940, 1982.
- [96] A. C. Ward, J. K. Liker, J. J. Cristiano, and D. K. Sobek II, “The second Toyota paradox: How delaying decisions can make better cars faster,” *Sloan Management Review*, vol. 36, no. 3, pp. 43–61, 1995.
- [97] D. K. Sobek II, A. C. Ward, and J. K. Liker, “Toyota’s principles of set-based concurrent engineering,” *Sloan Management Review*, vol. 40, no. 2, pp. 67–83, 1999.
- [98] A. M. Ross and D. E. Hastings, “The tradespace exploration paradigm,” *INCOSE International Symposium*, vol. 15, no. 1, pp. 1706–1718, 2005, doi:10.1002/j.2334-5837.2005.tb00783.x.
- [99] INCOSE, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed. Wiley, 2015.
- [100] D. L. Parnas, “On the design and development of program families,” *IEEE Transactions on Software Engineering*, vol. SE-2, no. 1, pp. 1–9, 1976, doi:10.1109/TSE.1976.233797.
- [101] S. Kong, J. Xia, D. Ruelas-Petrisko, Z. D. Sisco, J. Balkind, and G. H. Smith, “Improving equality saturation for EDA via semantic e-graphs,” *Proceedings of the ACM on Programming Languages*, vol. 10, PLDI, article 221, 2026, doi:10.1145/3808299.
- [102] R. Bojanić and E. Seneta, “A unified theory of regularly varying sequences,” *Mathematische Zeitschrift*, vol. 134, pp. 91–106, 1973.
- [103] J. Galambos and E. Seneta, “Regularly varying sequences,” *Proceedings of the American Mathematical Society*, vol. 41, no. 1, pp. 110–116, 1973.
- [104] J. Hajnal, “Weak ergodicity in non-homogeneous Markov chains,” *Proceedings of the Cambridge Philosophical Society*, vol. 54, no. 2, pp. 233–246, 1958.
- [105] J. E. Cohen, “Ergodic theorems in demography,” *Bulletin of the American Mathematical Society*, vol. 1, no. 2, pp. 275–295, 1979.
- [106] E. Seneta, *Non-negative Matrices and Markov Chains*, revised ed. Springer, 2006.
- [107] D. J. Hartfiel, *Nonhomogeneous Matrix Products*. World Scientific, 2002.
- [108] S. Williams, A. Waterman, and D. Patterson, “Roofline: An insightful visual performance model for multicore architectures,” *Communications of the ACM*, vol. 52, no. 4, pp. 65–76, 2009.
- [109] J. Symons et al., “Stream: Layer-fused multi-core dataflow exploration for deep neural networks,” 2023.
- [110] Y. S. Shao et al., “Simba: Scaling deep-learning inference with multi-chip-module-based architecture,” in *Proceedings of MICRO*, 2019.
- [111] E. T. Kalns and L. M. Ni, “Processor mapping techniques toward efficient data redistribution,” *IEEE Transactions on Parallel and Distributed Systems*, 1995.
- [112] A. Schrijver, *Theory of Linear and Integer Programming*. Wiley, 1986.
- [113] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of Computer Computations*, pp. 85–103, 1972.
- [114] J. H. Davenport and J. Heintz, “Real quantifier elimination is doubly exponential,” *Journal of Symbolic Computation*, vol. 5, pp. 29–35, 1988.

Collective attribution, accountability, and AI research provenance

Fitih M. Cinnor initiated, directed, and led the research program; established its scientific objectives and evidentiary standards; selected and adjudicated the claims advanced here; and reviewed the supporting mathematics and computation. He is the named and accountable human author, responsible for the accuracy, integrity, scope, and correction of the work. Accountable correspondence: fcinnor@semiaifoundry.com.

semiAIfoundry Research, a research group of Semi AI Foundry LLC, is the collective scholarly attribution for the coordinated research effort. It recognizes integrated intellectual work across human, artificial, and hybrid forms of intelligence rather than masking those contributions behind an exclusively individual account. Collective attribution does not displace human accountability or create AI legal authorship, ownership, signatory authority, representation authority, or accountable-person status.

The research used a human-directed, orchestrated multi-agent process. AI systems made material contributions to candidate generation, mathematical derivation, theorem formalization, counterexample construction, numerical implementation, software testing, high-precision and interval-oriented verification, literature mapping, prior-art comparison, research-record synthesis, adversarial critique, visualization, drafting, editing, and publication engineering. OpenAI GPT-5.6 Pro receives explicit credit for cross-corpus synthesis, mathematical exploration, derivation checking, theorem formalization, counterexample search, software and verifier development, exposition, consistency review, and manuscript construction. Anthropic Claude systems, including Fable 5, receive explicit credit for mathematical exploration, numerical-instrument development, adversarial review, gap identification, and editorial critique. The Fable 5 review was an internal AI-assisted audit, not independent external peer review or third-party validation.

These credits record provenance, not scientific status. AI-generated material acquires no standing merely by being generated. Scientific standing comes from proofs, computations, verification controls, independent reproductions, source checks, and correction records. No AI system is an author, legal owner, signatory, authorized representative, or accountable person. No identification or credit implies that OpenAI, Anthropic, EPFL, an artifact source, or any other provider or institution owns, sponsors, approves, warrants, or endorses this work, or accepts responsibility for its claims. Third-party rights remain governed by the applicable notices. The research program received no external funding. The accountable human author declares no competing interests or conflicts of interest.

Copyright, licensing, and artifact availability

Copyright © 2026 Semi AI Foundry LLC. Except where otherwise noted, the manuscript and project-created figures in the exact versioned public payload are licensed under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>), with requested attribution to “Fitih M. Cinnor for semiAIfoundry Research.” Reusers must give appropriate attribution, provide the license link, and indicate changes; attribution does not imply endorsement. Project-created code is licensed under Apache-2.0. Third-party materials remain governed by their recorded upstream terms. The licensed payload is identified by its versioned source, results, artifacts, version number, public file index, root SHA-256 manifest, and archival deposit. These grants apply only to the files and version identified in that payload and do not extend to other releases or evolving research-program content.