Computation Without Extraction

A Treatise on Distributed Intelligence, Thermodynamic Work,
and the Long-Term Economics of Existing Infrastructure

Flyxion
Independent Researcher

November 2026

Preface

This monograph proposes an organizing principle for computational infrastructure. Instead of asking how the utilization of dedicated computing assets can be maximized, it asks what computations existing physical and technological systems can perform without compromising the reasons those systems exist. A computer, a furnace, an air conditioner, and a communications link each serve an independent purpose. Each also possesses residual capabilities that conventional accounting treats as idle or wasted. The hypothesis examined here is that those residual capabilities can be organized into a useful computational network under conditions the monograph tries to state, and that the obstacle which most distinguishes the proposal is institutional and not only technical: the tendency of performance metrics to become targets that distort the purposes they were introduced to measure.

Three forms of distributed calculation are distinguished throughout: conventional digital computation using existing hardware, thermally integrated digital computation, and physical analog computation using actual thermodynamic processes. Much of the individual machinery is established. Volunteer computing, computational space heating, physical reservoir computing, and waste heat reuse each have a literature of their own. The contribution attempted here is their integration under a common theory of bounded participation, long-horizon adaptation, and resistance to metric-driven resource escalation, together with an explicit empirical program by which the integration can fail. The proposal is treated throughout as a conditional research hypothesis and not as a demonstrated solution. The final chapter sorts the monograph's claims into those established within its models, those conditional on quantities still to be measured, and those that remain open.

Symbols are defined where they are introduced and are local to the chapter in which they appear. The letters r, s, q, ρ, and ϵ are each used for different quantities in different chapters, in every case with the meaning stated at the point of use.

Contents

Preface **ii**

I The Efficiency Trap **1**

1 The Misallocation of Intelligence **2**
1.1 Six Quantities That Are Not One 2
1.2 Productive and Wasteful Idleness 3
1.3 A Worked Comparison 4
1.4 Terms Used Throughout 4
1.5 The Plan of the Argument 5

2 Jacques Ellul and the Supremacy of Technique **6**
2.1 Efficiency and Legitimacy 6
2.2 Normative Authority Without Defense 7
2.3 Which Adapts to Which 7
2.4 Influence Without Instruction 7
2.5 A Reactionary Objection 8

3 Goodhart's Law and the Collapse of the Objective **11**
3.1 Origins and Variants 11
3.2 A Formal Statement 12
3.3 Reward Design and the Bitcoin Case 12
3.4 An Institutional Consequence 13

4 Bitcoin and Competitive Escalation **15**
4.1 What the Expenditure Buys 15
4.2 Why More Computation Does Not Mean More Service 15
4.3 Competitive and Productive Computation 16
4.4 Heat Recovery and the Retrospective Justification 16

II Three Forms of Distributed Calculation **17**

5 Residual Digital Computation **18**
5.1 Resource Descriptors . . . 18
5.2 Precedents . . . 19
5.3 Availability and Task Granularity . . . 19
5.4 Matching Tasks to Nodes . . . 20
5.5 What Spare Memory Is and Is Not Good For . . . 20
5.6 Which Workloads Fit . . . 20
5.7 Conditions Under Which the Proposal Fails . . . 21

6 The Idea Router and Computation Without Execution **22**
6.1 Calculating and Enabling . . . 22
6.2 Descriptions, References, and Discovery . . . 22
6.3 Routing Reliability . . . 23
6.4 Matching and Its Limits . . . 24
6.5 Preserving Negative Results . . . 24
6.6 Recording and Authority . . . 24
6.7 Identity and Attack . . . 25
6.8 Conditions Under Which the Proposal Fails . . . 25

7 Thermally Integrated Digital Computation **26**
7.1 Thermodynamic Accounting . . . 26
7.2 The Heat-Pump Comparison . . . 27
7.3 An Admissibility Condition . . . 27
7.4 Storage and Timing . . . 28
7.5 Attributable Electricity per Unit of Computation . . . 28
7.6 Hardware Lifetime and Modularity . . . 29
7.7 Safety, Authority, and Deployment . . . 29
7.8 Conditions Under Which the Proposal Fails . . . 30

8 Physical Analog Computation **31**
8.1 Precedents . . . 31
8.2 A Formal Description . . . 31
8.3 Computation, Experiment, and Noise . . . 32
8.4 Similarity and Scaling . . . 33
8.5 When Physical Computation Is Cheaper . . . 33
8.6 Calibration, Validation, and Reproducibility . . . 34
8.7 Conditions Under Which the Proposal Fails . . . 34

III Thermodynamic Infrastructure as an Experimental Platform 35

9 Furnaces as Distributed Experimental Instruments 36

9.1 What a Residential System Can Test . . . 36
9.2 Similarity and Scaling . . . 37
9.3 Measurement and Uncertainty . . . 37
9.4 Site Effects and Identifiability . . . 38
9.5 When the Network Beats the Laboratory . . . 38
9.6 Safety and Certification . . . 39
9.7 A Staged Test and Its Failure Conditions . . . 39

10 Evolutionary Flow Geometry 40

10.1 Geometry and the Admissible Set . . . 40
10.2 Surrogates and the Information Objective . . . 40
10.3 The Evaluation Budget . . . 41
10.4 Selection Under Noise . . . 41
10.5 Safe Exploration . . . 42
10.6 Comparison with Numerical Optimization . . . 43
10.7 A Distributed Dataset . . . 43
10.8 Conditions Under Which the Proposal Fails . . . 44

11 Refrigeration as Physical Computers 45

11.1 The Cycle as a Physical System . . . 45
11.2 What Question Can the Loop Answer . . . 46
11.3 Fault Detection Across a Fleet . . . 46
11.4 Non-Invasive Instrumentation . . . 47
11.5 Admissibility and the Thermostat . . . 47
11.6 Seasonal Complementarity . . . 47
11.7 Conditions Under Which the Proposal Fails . . . 48

12 Entropy as a Service 49

12.1 Four Notions of Entropy . . . 49
12.2 Characterizing a Physical Source . . . 49
12.3 Extraction and Its Limits . . . 50
12.4 Combining Many Untrusted Sources . . . 51
12.5 Is the Service Worth Providing . . . 51
12.6 An Admissible Entropy Contribution . . . 52
12.7 Conditions Under Which the Proposal Fails . . . 52

IV The Economics of Existing Things 53

13 Sunk Infrastructure and Marginal Computation 54
13.1 Three Kinds of Cost . . . 54
13.2 The Comparison Condition . . . 54
13.3 Operating Energy and the Strongest Counterexample . . . 55
13.4 Cost of Participation and Cost of Organizing Participation . . . 56
13.5 Rebound and Demand . . . 57
13.6 When Reuse Is Counterproductive . . . 57
13.7 Conditions Under Which the Economic Case Fails . . . 58

14 The Geography of Computational Demand 59
14.1 Which Work Can Move . . . 59
14.2 Where the Heat Is Needed . . . 60
14.3 The Grid Behind the Heater . . . 61
14.4 Jurisdiction and Connectivity . . . 61
14.5 Placement as an Optimization . . . 61
14.6 Who Bears the Placement . . . 62
14.7 Conditions Under Which the Proposal Fails . . . 62

15 The Long Horizon of Efficiency 63
15.1 Instantaneous and Lifetime Efficiency . . . 63
15.2 History as a Productive Resource . . . 64
15.3 A Learning-Curve Model . . . 64
15.4 Irreversibility and the Option to Stop . . . 66
15.5 Falsifiability and Experiment D . . . 67

V Governance, Goodhart, and Selective Continuation 68

16 Not Rewarding Raw Computation 69
16.1 A Taxonomy of Gaming . . . 69
16.2 Verification . . . 70
16.3 Bounding Participation . . . 71
16.4 Gates and Fines . . . 72
16.5 Provenance, Independence, and Review . . . 72
16.6 The Governing Rule and Its Limits . . . 72

17 Local Authority and the Right to Remain Idle 74
17.1 A Hierarchy of Authority . . . 74
17.2 Revocation . . . 75
17.3 Local Priority . . . 76
17.4 Privacy and Isolation . . . 76

17.5 Availability Is Not Authorization 76
17.6 How the Right Erodes 77
17.7 Dependence and Reliance 77
17.8 Conditions Under Which the Principle Fails 78

18 An Ecology of Admissible Contributions 79
18.1 The Three Forms and the Unit of Participation 79
18.2 Results Established Within the Models 80
18.3 Conditional Propositions 81
18.4 Open Conjectures 81
18.5 Status of the Opening Promises 82
18.6 Order of Operations 82
18.7 Closing Statement 83

A Mathematical Foundations 85

B Empirical Program 87
B.1 Principles Common to the Experiments 87
B.2 Experiment A: Residual Computing 88
B.3 Experiment B: Computational Heating 89
B.4 Experiment C: Physical Analog Calculation 91
B.5 Experiment D: Long-Horizon Evolutionary Scheduling 93
B.6 Experiment E: Goodhart Stress Testing 94
B.7 Summary of the Program 96

Part I

The Efficiency Trap

Chapter 1

The Misallocation of Intelligence

The contemporary debate over centralized artificial intelligence infrastructure frequently rests on a premise that is seldom stated outright: that a scarce computational resource is used well to the extent that it is used continuously. On this premise, memory installed in personal computers is a stranded asset whenever it is not serving a request, and the natural remedy is to aggregate it, schedule it, and bring its utilization closer to that of a purpose-built facility. The argument was stated plainly by Sam Altman in an October 2026 podcast appearance, as reported by Bonifield [8]. Asked about the ongoing shortage of DRAM, Altman said that from a global allocation perspective it is quite terrible for everyone to hold a share of the world's limited memory supply in a personal computer where it is used five percent of the time, and that maximum utilization in the cloud is the right aim in a supply-constrained environment. The argument has real force. Hardware that is manufactured and then left unused represents embodied energy and material that produced no service.

The question this chapter poses is whether maximizing the utilization of a scarce resource is equivalent to maximizing its social value. The two coincide only under conditions that rarely hold together: that the resource has a single use, that its idle time has no value of its own, that the cost of coordinating its use is negligible, and that the measure of utilization tracks the purpose for which the resource exists. Where any of these fails, a high utilization figure and a high social value can diverge, and the divergence is invisible to anyone who consults only the figure.

1.1 Six Quantities That Are Not One

The vocabulary of efficiency conceals several distinct quantities. Instantaneous utilization is the fraction of capacity engaged at a given moment. Lifetime utilization is the fraction of capacity engaged over the service life of the hardware, which can be high even when instantaneous utilization is usually low, provided the bursts are long enough. Economic efficiency compares the value of outputs to the value of all inputs, and depends on prices

that are themselves shaped by institutional arrangements. Energy efficiency compares useful output to energy consumed, and says nothing about the energy embodied in manufacture. Resilience is the ability to continue to provide a service under disturbance, and it is typically purchased with redundancy that utilization metrics register as waste. Computational autonomy is the ability of an owner to run calculations without depending on a third party's network, terms, or continued existence.

These six quantities cannot be collapsed into a single objective without choosing weights, and the choice of weights is a value judgment rather than a technical result. A system tuned for instantaneous utilization will tend to sacrifice resilience and autonomy, because both require capacity held in reserve. A system tuned for autonomy will tend to show poor utilization, because the owner's machine must be able to serve the owner at any moment. The tension between them is structural. It is not a transitional inefficiency awaiting a better algorithm.

1.2 Productive and Wasteful Idleness

A computer that is idle for most of the day may still be doing something valuable: it is available. Availability has a price in any market that sells it. A standby generator, a fire engine, and a spare tire are all idle almost all of their lives, and no one describes their idleness as a failure of allocation. Their value lies in the option they confer, the ability to act immediately and locally when need arises.

This motivates a distinction between productive idleness and wasteful idleness. Idleness is productive when unused capacity functions as insurance against future demand, as protection of privacy or local control, or as headroom that keeps the primary function responsive. It is wasteful when no such function is served and the capacity could be directed elsewhere without cost to the owner. The difficulty is that the two cannot be told apart from a utilization trace alone. Both appear as a flat line near zero. Distinguishing them requires knowing what the idle capacity is for, which is information that resides with the owner and not with the scheduler.

Definition 1.1 (Residual capability)**.** Let a system S serve a primary purpose P. A *residual capability* of S is a computational or physical action that S can perform without degrading its performance of P below a level specified independently of any use of the residual capability.

The qualifier "specified independently" carries the weight of the definition. If the acceptable level of degradation is set by the party that wishes to use the residual capability, the definition becomes circular, and any capability whatsoever can be called residual by choosing a sufficiently tolerant standard. The same problem recurs in Chapter 3 as the general difficulty of metrics that are controlled by those they measure, and in Chapter 17 as the governance question of who decides what a system may spare.

1.3 A Worked Comparison

The inference in the utilization argument has three steps, and they are not equally secure. Higher utilization is taken to imply better allocation, and better allocation is taken to imply the preferred social arrangement. The first step requires empirical assumptions about workload, memory per unit of workload, and the cost of whatever infrastructure delivers the service. The second requires a normative argument that a utilization figure cannot supply, because it concerns who controls the resource and on what terms.

The first step can be made explicit. Let W denote workload demand, m the memory required per unit of workload, u the utilization achieved, and I the additional infrastructure needed to deliver the service, all expressed in comparable resource units. Writing subscript c for centralized provision and l for local provision, total resource consumption is approximately

$$R_c = \frac{W_c\, m_c}{u_c} + I_c, \qquad R_l = \frac{W_l\, m_l}{u_l} + I_l.$$

The condition $u_c > u_l$ does not establish $R_c < R_l$. Centralization adds networking, redundancy, cooling, security, and administration to I_c. It can also raise W_c, since a service that is easier to reach tends to be used more, so that the savings from higher utilization are partly or wholly absorbed by increased demand. The comparison that matters is the total resource consumed to deliver an equivalent level of service, including reliability and latency, and a single utilization figure leaves that comparison open.

Two further observations bear on how the argument should be weighed. The reported figure of five percent is given without a stated methodology, and memory occupancy, memory bandwidth, and computational utilization are different quantities. And the scarcity to which the argument responds is partly the product of the same allocation. The report states that large memory manufacturers hold long-term agreements with cloud operators and with projects such as Stargate that have reserved much of the world's output for the next several years [8]. A scarcity shaped by the purchases of the parties advocating centralization cannot simply be treated as an external constraint, and those parties also benefit commercially if consumers shift toward cloud inference. This does not make the engineering observation false. It means that a proposal to allocate scarce resources toward the proposer's business model should be assessed on the total-resource comparison, not on the proposer's chosen metric.

1.4 Terms Used Throughout

Four terms recur and are kept distinct, because each can hold while the next fails. Available capacity is what a device can physically do beyond its present load. It describes hardware and implies no permission. A residual capability, in the sense of Definition 1.1, is the narrower notion of an action the device can perform without degrading its primary function below a level specified independently of the use. Authorized capacity is the part of that

which the owner permits and the safety envelope allows, as developed in Chapter 17. Useful computation is computation whose results have independently verified value, as developed in Chapter 16. A contribution is admissible, in the sense of Definition 18.1, only when it is residual, authorized, verified, and valuable. A resource can be physically idle without being authorized. An authorized contribution can still be economically disadvantageous. Completed computation can lack independently verified value. A statement in a later chapter that uses one of these terms should be read with its own meaning, and a result about one is not a result about the others.

1.5 The Plan of the Argument

The remaining chapters of Part I develop the philosophical and institutional background to this problem. Chapter 2 examines the claim that the pursuit of efficiency, once elevated to a supreme criterion, subordinates the purposes it was meant to serve. Chapter 3 supplies a formal account of how a measurable proxy can diverge from the objective it stands for under optimization pressure. Chapter 4 considers proof-of-work mining as the clearest existing case of a reward structure that converts computational expenditure into an end.

Part II then turns to an alternative organization of computation in which the unit of account is not utilization but the admissible use of capacity that already serves another purpose. The alternative may prove less efficient at first, and it may prove less efficient throughout. The monograph treats it as a research hypothesis whose case, if it can be made, rests on avoided capital, local authority, and learning over a long horizon. The later chapters test the hypothesis against findings that cut the other way. Centralized operators can learn faster in calendar time (Chapter 15). Distributed capacity is geographically correlated (Chapter 14). Displacing heat pumps can raise peak electricity demand (Chapter 14). Existing hardware is not free to operate (Chapter 13). And the alternative is only partly resistant to the failure mode that Part I describes (Chapter 16).

Chapter 2

Jacques Ellul and the Supremacy of Technique

The philosophical foundation of this monograph is taken from Jacques Ellul's *The Technological Society* [15]. Ellul's central concept, technique, does not refer to machines. It refers to the totality of methods rationally arrived at and having absolute efficiency, in every field of human activity, for a given stage of development. A machine is one embodiment of technique. A bureaucratic procedure, an advertising campaign, and an accounting convention are others. What they share is that each is chosen because it is the most efficient available means.

The argument examined here is not that efficiency is undesirable. Ellul's claim is narrower and more demanding: that when efficiency becomes the supreme criterion, competing values are subordinated to it without ever being explicitly defeated in argument. A choice that is technically superior needs no further defense, and a choice that is technically inferior must justify itself against a standard it did not set. Over time the burden of proof shifts so completely that alternatives are rejected for failing a test that no one chose to apply.

2.1 Efficiency and Legitimacy

The chapter turns on a distinction between two propositions that ordinary speech runs together. The first is that an arrangement is efficient, meaning that it achieves a stated result at low cost. The second is that the arrangement is legitimate, meaning that it can be justified to the people it affects, with respect to ends they have reason to accept. The first is a technical claim, answerable to measurement. The second is a normative claim, answerable to argument. A very efficient arrangement can be illegitimate, as when it achieves a result at low cost by imposing the cost on parties who never consented. A legitimate arrangement can be inefficient, as when a community chooses a slower process because it values participation.

Technique, in Ellul's account, collapses the second proposition into the first. Because efficiency can be measured and legitimacy cannot, the measurable quantity comes to stand in for the whole. This is the point at which the argument of this monograph meets the

argument of Chapter 3. The elevation of efficiency to a supreme criterion is a special case of a measurable proxy acquiring authority over the purpose it was introduced to serve.

2.2 Normative Authority Without Defense

One of the more subtle observations in this tradition is that technical optimization can acquire normative authority without anyone defending the social arrangement it produces. No one need argue that a city should be organized around the automobile. The organization follows from a sequence of locally efficient decisions, each defensible, none of them a decision about the city. The aggregate arrangement is then treated as a given to which individuals must adapt.

Computational infrastructure exhibits the same pattern. A series of reasonable decisions to consolidate hardware, to raise utilization, and to amortize fixed costs over larger workloads yields an arrangement in which computation is concentrated in a small number of facilities and everyone else is a client. No single decision was a decision to centralize intelligence. The centralization is an emergent consequence, and its justification, where one is offered, is that it is the efficient outcome.

2.3 Which Adapts to Which

The question that organizes the rest of the monograph can now be stated. Should computational infrastructure determine the organization of human activity, or should it adapt to purposes established independently of it? The first answer treats the infrastructure as fixed and fits households, buildings, and institutions to it. The second treats the purposes as fixed, among them keeping a building warm, keeping food cold, and preserving a person's control over their own machine, and asks what computation can be done within them.

The second answer does not reject efficiency. It places efficiency in a subordinate position: among the ways of serving a purpose, choose the efficient one, but do not let the pursuit of efficiency redefine the purpose. This ordering is the sense in which the proposed architecture is a response to Ellul, and Chapter 17 returns to it as a principle of governance.

2.4 Influence Without Instruction

A closely related mechanism appears in an unrelated field and clarifies what is at stake. In a 2026 essay on technology journalism, Caffier [11] describes how corporate hospitality, including sponsored travel, managed tours, and gifts, can shape reporting without any request for favorable coverage. His point is that the reporter may truthfully state that no promises were made and that the sponsorship was disclosed, and still find that the access altered which questions seemed natural to ask and which stories seemed worth pursuing. A bribe tries to change a decision. Managed access changes the set of decisions that are

psychologically and professionally available, and the stories that are never pursued do not appear as editorial choices that anyone could audit.

The mechanism can be stated compactly. Let H denote the investigator's evidence, A the access conditions established by the subject, and Q the question the investigator pursues. Editorial independence guarantees that the subject cannot dictate the published conclusion. It does not guarantee

$$P(Q \mid H, A) = P(Q \mid H),$$

and the purpose of a sophisticated public relations operation may be precisely to make these distributions differ without issuing any instruction. Caffier's essay is a single first-person account, and its broader claims about the industry and about the mental states of the people he meets are not demonstrated by the encounters he describes. It is used here only to illustrate a mechanism, which parallels the one in this chapter: an optimized arrangement can acquire authority through the conditions it creates and not through any argument it makes.

Ellul's distinction is the useful one. An arrangement can be genuinely efficient at what it does, whether the arrangement is centralized hardware utilization or hospitality as a communication technique, and that achievement does not establish that the resulting distribution of knowledge or control is desirable. Even if the party optimizing were benevolent and its calculations entirely accurate, the underlying problem would remain. The question is not only whether the optimization works, but who decided that the thing being optimized should govern everything else.

2.5 A Reactionary Objection

The Colombian aphorist Nicolás Gómez Dávila supplies an objection that is more severe than Ellul's, and the monograph should answer it even though it does not require a chapter of its own. His scholia [17], brief observations written in a reactionary and Catholic mode, doubt that historical novelty, democratization, and growing technical capability constitute improvement, and the doubt extends to the modern faith in deliberate social reconstruction. His position should not be conflated with Ellul's. Ellul examines how efficiency, once supreme, displaces other forms of judgment. Gómez Dávila rejects the more general identification of the new with the better. Nor can he be recruited as a patron of distributed computing, since his skepticism of engineered social arrangements makes him more useful here as a critic of the project than as an authority for it. The objection below is constructed for the monograph from the stance he is known for, and it is not a claim about what he said of computers.

The objection is that an alternative technological system may reproduce the assumptions of the system it opposes in a different organizational form. The proposed architecture is itself designed, specified, and quantified. Its vocabulary of residual capacity treats the domestic sphere as a field of resources waiting to be organized, and it presupposes that

computation discovered anywhere is worth seeking. A network of reused machines may look backward by conventional measures of performance and still be advantageous in resilience, autonomy, reversibility, and avoided capital. Those criteria are also measures, however, and they are exposed to the same corruption as the proxies of Chapter 3. The distinction between progress as increasing capability and progress as preserving the conditions under which people can live well does not escape the difficulty by being drawn, because the second kind of progress can be turned into a metric and optimized just like the first.

The monograph's reply is partial. It does not claim to escape technique. It claims, in the remark that follows, that the order of priority between a purpose and its measure can be made explicit in the design of an institution. The architecture also carries limits on itself: the right to remain idle, opt-in participation, consent that expires, and stated conditions under which it fails. The one feature that answers the objection on its own terms is reversibility, since a system that can be abandoned without harm has not captured those who joined it. But reversibility asserted as an intention is not reversibility. Organizational commitments accumulate, as Chapter 15 observes, and dependence grows through subsidies and bundled services, as Chapter 17 observes. A test is needed that does not rely on good intentions.

Definition 2.1 (Dissolution condition)**.** A network satisfies the *dissolution condition* at time t_0 if its dissolution at t_0 would meet three requirements. First, functional restoration: each participating system performs its primary function no worse than before it joined. Second, financial restoration: every cost that a participant incurred solely because of participation, and has not recovered through the network's benefits, is borne by the network or its operator from a reserve held for the purpose. Third, release: no obligation to the network survives.

The condition is demanding, and it is not met by default. A household that removed its conventional heater to install a computing module, or whose heating subsidy depends on continued participation, or whose hardware cannot be recovered, fails it. Read literally, full restoration is impossible, since participants spend time and attention that no reserve can return, and a participant who has come to rely on the network's services cannot be made as though the network had never existed. The condition is therefore stated in three parts that can be checked, and it asks for restoration of function and of identifiable costs, not of everything. It also separates two kinds of dependence. Dependence of a host's primary function on the network, as when a household loses heat when the network is disconnected, is prohibited, and the resistance fallback element of Chapter 7 is the means of preventing it. Ordinary voluntary reliance on a service that exists only because the network does is permitted, since a network that provides nothing worth relying on has no case for existing.

The second requirement can be made quantitative. If $U_i(t)$ denotes the unrecovered participation cost of participant i at time t, the network must hold a dissolution reserve $R(t)$ satisfying

$$R(t) \;\geq\; \sum_{i} U_i(t),$$

which places the cost of withdrawal on the network and not on its participants. The reserve

is itself an organizational commitment of the kind identified in Chapter 15, and it reduces the network's attractiveness at small scale by the amount it ties up. Modular, retrievable hardware reduces each U_i directly, and the no-conditionality rule of Chapter 17 keeps subsidy dependence from arising. A tension remains that no rule removes. The stronger the services of the network become, the more reasonably its participants will rely on them, and the greater the pressure to preserve the network for their sake. The condition cannot guarantee that nobody comes to depend on the network while the network also grows into substantial infrastructure. What it can do is keep dependence for essentials absent and make the cost of leaving explicit, so that the momentum of a useful network is not purchased with the captivity of its participants. The condition is necessary for the architecture to count as reversible and it is not sufficient for it to count as legitimate. A system can satisfy it and still reproduce the assumption that idle things exist to be organized, and the remainder of the objection is left open.

Remark 2.2. Ellul's pessimism about the autonomy of technique should be kept distinct from the engineering claims of later chapters. The proposal here is not that technique can be abolished or that its logic can be ignored. It is the more modest claim that the order of priority between a purpose and its measure can be made explicit in the design of an institution, and that doing so changes what the institution rewards.

Chapter 3

Goodhart's Law and the Collapse of the Objective

This chapter is the theoretical bottleneck of the monograph. Every later claim about reward mechanisms, verification, and long-horizon accounting depends on a precise account of how a measure comes apart from the purpose it measures.

3.1 Origins and Variants

The law that bears Goodhart's name originates in a 1975 discussion of British monetary policy [18], where the observation was that a statistical regularity tends to collapse once pressure is placed upon it for control purposes. It has since been compressed into the slogan that a measure ceases to be a good measure when it becomes a target, a phrasing usually traced to Marilyn Strathern's discussion of audit in British universities [37]. The compression is convenient and slightly misleading, because it merges several distinct claims.

Donald Campbell stated a closely related principle for social indicators [12]: the more a quantitative indicator is used for social decision making, the more subject it becomes to corruption pressures and the more apt it is to distort the processes it is intended to monitor. Campbell's formulation locates the failure in the behavior of people under evaluation. Goodhart's original formulation locates it in the breakdown of a statistical relationship. Later work in machine learning, under the names reward hacking and specification gaming [35], locates it in the gap between a specified reward and the intended behavior. Manheim and Garrabrant [29] separate four mechanisms by which the divergence can arise: regressional, extremal, causal, and adversarial. The distinction matters because the remedies differ. Regressional divergence is addressed by better estimation, and adversarial divergence only by changing the incentives of the parties who benefit from the gap.

3.2 A Formal Statement

Let $W(x)$ denote the intended social objective as a function of an action or configuration x, and let $M(x)$ denote the measurable proxy that an institution actually optimizes. The proxy is introduced because W is not directly observable or is too costly to measure. The difficulty arises when

$$\arg\max_x M(x) \neq \arg\max_x W(x).$$

Computational throughput, utilization, benchmark performance, transactions processed, and revenue generated are all candidate proxies in the settings considered here. Once an institution optimizes such a measure, it can produce impressive numerical results without achieving the purposes that the measure was meant to represent.

A minimal model shows how optimization pressure alone can produce the divergence. Suppose a participant holds a budget $B > 0$ and divides it between productive effort $p \geq 0$ and gaming effort $g \geq 0$, so that $p + g = B$. Productive effort contributes to both the proxy and the objective. Gaming effort contributes to the proxy at rate $a > 0$ and detracts from the objective at rate $b \geq 0$ through the resources it diverts.

$$M(p, g) = p + a\,g, \qquad W(p, g) = p - b\,g.$$

Proposition 3.1. *If $a > 1$, the participant who maximizes M sets $g = B$ and obtains $W = -bB \leq 0$, whereas the allocation that maximizes W sets $g = 0$ and obtains $W = B > 0$. If $a \leq 1$, the maximizer of M also sets $g = 0$ and the proxy is faithful at its optimum.*

Proof. Substituting $p = B - g$ gives $M = B + (a - 1)g$ and $W = B - (1 + b)g$ on $0 \leq g \leq B$. The function W is strictly decreasing in g for every $b \geq 0$, so its unique maximizer is $g = 0$ with value B. The function M is strictly increasing in g when $a > 1$, so its maximizer is $g = B$, where $W = B - (1 + b)B = -bB$. When $a \leq 1$, M is nonincreasing in g and $g = 0$ is a maximizer. □

The model is deliberately crude and is offered as an illustration of a mechanism, not as a description of any particular system. It makes one point that the later chapters rely on: the proxy can be perfectly well behaved over the range of ordinary behavior and still fail at its optimum, because the optimum is exactly where the cheapest way of raising the proxy is exploited. The threshold $a > 1$ says that the proxy fails as soon as gaming is more rewarded than production, and it fails completely and not gradually.

3.3 Reward Design and the Bitcoin Case

The model of the preceding section concerns an individual who splits a fixed budget. Reward design raises a second question, which is what a reward proportional to measured activity does to the total expenditure of a population. Consider $N \geq 2$ identical participants who each choose a hardware expenditure $x_i \geq 0$ at unit cost, and suppose a reward pool of value

R is divided in proportion to expenditure, so that participant i receives $R\,x_i/\sum_j x_j$. This is the structure of a hash race, and of any scheme that pays for raw compute contributed.

Proposition 3.2. *In the symmetric equilibrium of this contest each participant spends $x^* = R(N-1)/N^2$, so that total expenditure is $R(N-1)/N$, which approaches R as N grows, and each participant's net payoff is R/N^2.*

Proof. Participant i maximizes $\pi_i = R\,x_i/(x_i + X_{-i}) - x_i$, where $X_{-i} = \sum_{j\neq i} x_j$. The first-order condition is $R\,X_{-i}/(x_i+X_{-i})^2 = 1$. At a symmetric point $x_i = x$ and $X_{-i} = (N-1)x$, which gives $R(N-1)x/(Nx)^2 = 1$ and hence $x = R(N-1)/N^2$. Total expenditure is $Nx = R(N-1)/N$, and the net payoff is $R/N - x = R/N^2$. □

The result is a standard one in the theory of contests, and its relevance here is what it implies about a computation reward. When participants are many and alike, nearly the whole value of the reward is dissipated in expenditure, and the expenditure is a transfer to suppliers of hardware and electricity that adds nothing to the stock of independently useful results. Doubling the number of participants does not enlarge the reward and only raises the competitive spending that the reward induces. This is the formal counterpart of the escalation described in Chapter 4, and it explains why a network that pays by quantity of compute inherits the same behavior regardless of the worthiness of the tasks. In the notation of Proposition 3.1, the proxy is expenditure, the objective is useful output, and the reward makes expenditure the cheapest route to income.

The remedy that Chapter 16 develops is to attach reward to verified outcomes and not to activity. That shifts the contest without ending it. Once results are what is paid for, the cheapest way to raise income becomes fabrication, duplication of trivial tasks, or manipulation of the baseline against which savings are measured, and the validation procedure becomes the new object of gaming. No reward rule removes the pressure. What can be done is to bound participation, so that the maximum useful expenditure of a participant is limited by the purpose of the host system, and to verify results independently of those who profit from them.

3.4 An Institutional Consequence

The thesis of this chapter is that Goodhart-type failure is not merely an occasional accident of optimization. It tends to recur when a measurable proxy is allowed to acquire authority over the underlying purpose, and three features make the recurrence likely, though not inevitable. First, the objective W is almost always harder to observe than the proxy M, which is why the proxy was adopted. Second, the party whose behavior is measured typically has better information about how to move M than the party that designed it. Third, rewards attached to M create a standing incentive to find the cheapest means of increasing it.

Taken together, these features suggest that a stable institution should not rely on the goodwill of participants to keep M and W aligned. Alignment must be maintained

structurally, by constraining which actions are admissible regardless of their effect on M. This is the move taken in Part V: the network may not reward a participant for defeating the constraints that justified the participant's inclusion, and the verification of results is separated from the measurement of activity.

Chapter 4

Bitcoin and the Competitive Escalation of Computation

Proof-of-work mining supplies the monograph's principal counterexample, a system in which a reward structure converts computational expenditure into a goal in its own right. It is a fair choice because the system is not frivolous. It was designed to solve a real problem, and any argument about its costs has to begin by acknowledging what the expenditure buys.

4.1 What the Expenditure Buys

In the design described by Nakamoto [31], participants compete to find a value whose hash satisfies a difficulty condition, and the winner of each round is entitled to append a block of transactions to a shared ledger and collect a reward. Because finding such a value requires repeated trial, a participant who wishes to rewrite the ledger's history would have to redo the work from the point of the alteration forward, and would have to do it faster than the honest participants combined. The expenditure is therefore not computationally meaningless. It makes reorganization of the ledger costly, and that cost is the system's security function.

4.2 Why More Computation Does Not Mean More Service

The network adjusts its difficulty parameter at fixed intervals, at roughly every two thousand blocks, so that blocks continue to arrive at a target average rate. The consequence is that an increase in aggregate computational effort does not produce a proportional increase in the rate of confirmed blocks, or in the transactional capacity that users receive. The additional effort raises the difficulty. What it raises in return is the cost an attacker would face, together with the cost every participant bears to remain competitive.

The chapter must therefore keep two claims separate. The first is that proof-of-work provides security by making ledger reorganization costly, which is true by design. The second is that every additional unit of computation creates additional social value, which does not

follow from the first. Whether a given level of expenditure is the right price for a given level of security is a question about marginal benefit, and the reward structure does not ask it. The reward structure pays for being first, not for being useful.

4.3 Competitive and Productive Computation

The general distinction drawn from this case is between two kinds of computational activity.

Definition 4.1 (Competitive and productive computation)**.** Computation is *competitive* when participants expend resources to improve their position relative to other participants, so that the benefit of additional expenditure to one participant is partly a loss to others. Computation is *productive* when additional resources produce results that have independent value whether or not other participants also compute.

A hash race is competitive in this sense. The benefit to a miner of doubling hardware is largely a reduction in everyone else's share. A protein-folding simulation is productive, since a second simulation of a different protein is valuable regardless of whether anyone else has run the first. Competitive computation exhibits escalation: when a reward is divisible among participants in proportion to expenditure, each has an incentive to spend more, and the equilibrium level of expenditure tends toward the value of the reward. The expenditure that results is a transfer to the suppliers of hardware and electricity, and not an addition to the stock of useful results.

4.4 Heat Recovery and the Retrospective Justification

Some mining operations reuse the heat their machines produce, for greenhouses, buildings, or industrial drying. The chapter distinguishes two situations that look alike from outside. In genuine thermal substitution, the heat displaces fuel or electricity that would otherwise have been consumed for heating, so the computation rides on an energy demand that already existed. In retrospective justification, an operation is built for the sake of its computational reward, and heat recovery is added afterward to defend the electricity consumption it created. The first reduces net demand relative to the alternative. The second increases net demand and recovers part of it.

The test that separates the two cases is a counterfactual one: would the heat have been needed, and the energy bought, without the computation? Only the first case passes. This test returns in Chapter 7 as the governing rule of thermally integrated computation, in which the heating requirement of a building sets the permissible computation, not the reverse.

Part II

Three Forms of Distributed Calculation

Chapter 5

Residual Digital Computation

The first form of distributed calculation uses existing computers as computational resources. Instead of specialized hardware or substantial dedicated installations, the network identifies the RAM, storage, processing capacity, and bandwidth that already exist on machines serving other purposes. Each participant sets a bounded contribution policy, for example five or ten percent of spare resources. The economic analysis of Chapter 13 needs an operationally precise account of what such a network can in fact contribute, and this chapter supplies it. The account is deliberately sober. Several of the things that spare capacity is imagined to be good for turn out to be beyond it.

5.1 Resource Descriptors

A node i advertises its state at time t as a descriptor

$$r_i(t) = \big(c_i(t),\, m_i(t),\, s_i(t),\, b_i(t),\, h_i(t),\, q_i(t)\big),$$

whose entries are the available compute, memory, storage, network capacity, useful thermal capacity, and reliability or availability characteristics. Tasks are assigned according to these capabilities, and nodes are not assumed to be interchangeable.

A local policy determines which of these resources the device may contribute. An owner might permit at most ten percent of spare computational capacity, five percent of available storage, and limited bandwidth. The percentages apply to the resources actually available after local requirements have been met, and not to nominal hardware capacity, which is never assumed to be disposable. An older machine with four gigabytes of memory might contribute a few hundred megabytes. Another might offer only spare storage, and a third a graphics processor for a few evenings a week. A fourth might have almost nothing to spare but a reliable network connection, which suits it to the coordination role described in Chapter 6.

Measuring what is spare is itself a design problem. Operating-system counters report processor utilization and free memory, but memory that is nominally free may be serving

as a cache that keeps the owner's applications responsive, and borrowing it can degrade the owner's experience in ways the counters do not show. The contribution policy should therefore be defined against an interference criterion, such as a bound on the increase in the owner's response latency, and the network's work must yield immediately when that bound is approached, in line with the priority rule of Chapter 17.

5.2 Precedents

The chapter compares the architecture with volunteer computing, including SETI@home [1] and the BOINC platform [2], with peer-to-peer systems, with edge computing, and with opportunistic scheduling. These systems establish that large heterogeneous populations of volunteer machines can complete useful work. They also establish the characteristic failure modes: unreliable nodes, falsified results, and participation that tracks credit rather than purpose, which is the subject of Chapter 16. The original contribution here lies in treating limited participation and hardware heterogeneity as foundational design requirements and not as temporary deficiencies. A scheduler that assumes a large and growing pool of identical contributors will be disappointed by a population in which each contributor spares a small fraction of a modest machine. A scheduler designed from the outset for small, uneven, revocable contributions has a different structure: short tasks, verifiable results, tolerance of departure, and no dependence on any single node.

5.3 Availability and Task Granularity

Spare capacity is intermittent, and the length of a task must be matched to the length of the intervals in which a node is available. Model the durations of a node's availability sessions as independent and exponentially distributed with rate λ, so that the mean session is $1/\lambda$. Suppose a task needs an uninterrupted period d and cannot be resumed.

Proposition 5.1. *Under these assumptions each attempt succeeds with probability $e^{-\lambda d}$, and the expected number of attempts needed to complete the task is $e^{\lambda d}$.*

Proof. An attempt succeeds exactly when the session lasts at least d, which has probability $\Pr[S \geq d] = e^{-\lambda d}$ for a session length S with rate λ. Attempts use independent sessions, so the number of attempts is geometric with success probability $e^{-\lambda d}$, whose mean is $e^{\lambda d}$. $\square$

The expected cost grows exponentially in the ratio of task length to mean session length, so a task that is twice as long as the mean session needs on average more than seven attempts. Tasks that cannot be checkpointed should therefore be kept short relative to $1/\lambda$, for example with $d \leq (\ln 2)/\lambda$ so that the expected number of attempts is at most two, and longer tasks need to checkpoint and resume, which adds overhead and requires storage. The exponential model is a simplification, since real availability is shaped by daily and weekly routines, and the descriptor's reliability entry q_i is meant to carry an empirical estimate in its place.

5.4 Matching Tasks to Nodes

Given tasks with requirements $(c_j, m_j, s_j, b_j, d_j)$ and nodes with descriptors, a task can be assigned to a node only if the node's current resources meet the requirements and its expected availability covers d_j. This defines a compatibility graph between tasks and nodes. If each node runs at most one task at a time and each task is assigned at most once, the best assignment is a maximum-weight matching in a bipartite graph, which can be found in polynomial time [25]. Real scheduling is harder, because nodes can host several tasks with different resource demands, which is a vector packing problem that is computationally intractable in general, and in practice the network will rely on heuristics and on the history of Chapter 15 to improve them. The matching formulation is useful as a reference point: it identifies the best the scheduler could do with perfect information, against which its performance can be measured.

5.5 What Spare Memory Is and Is Not Good For

The argument that motivated Chapter 1 concerned memory, and the question of what a network of spare memory can do deserves a direct answer. The answer is less than the aggregate capacity suggests. The memory of a single machine is accessed at bandwidths of tens of gigabytes per second, whereas a household network connection delivers on the order of ten megabytes per second, a ratio of roughly three orders of magnitude. A model whose parameters are spread across many machines connected by such links would spend nearly all its time waiting for data. Residual memory scattered across a wide-area network is therefore not a substitute for the high-bandwidth memory of a dedicated accelerator, and it cannot be used to serve large models in interactive settings.

What residual memory and storage can do is hold content that is read rarely or in large sequential pieces: replicas of data sets, cached results, shards of archives, and the records of the history of Chapter 15, addressed by content as described in Chapter 6. The network is a good place to keep data that must persist and be available somewhere, and a poor place to keep data that must be reached quickly and often.

5.6 Which Workloads Fit

The deciding quantity for a wide-area task is the ratio of computation to communication. If a task needs D bytes moved over a link of bandwidth B and requires compute time T_c, then with no overlap of transfer and computation the fraction of time spent usefully is

$$\eta_d = \frac{T_c}{T_c + D/B}.$$

A task with ten megabytes of input over a link carrying ten megabytes per second spends one second moving data, so it needs at least about ten seconds of computation to keep

the overhead near ten percent. The workloads that fit are those with large computation per byte and little coupling between tasks: Monte Carlo simulation, parameter sweeps, search over large spaces, rendering in independent tiles, and batch analysis of independent records. Workloads that exchange data continuously between tasks, or that need results within milliseconds, do not fit, and the network should not be offered for them.

5.7 Conditions Under Which the Proposal Fails

The proposal fails when the work that fits the network’s constraints is not valuable enough to cover its coordination and replication costs, when measured interference with the owners exceeds the bound they have set, when availability is too short or too erratic for tasks of useful length, or when the share of spare resources that owners are willing to commit is too small for the network to reach a useful scale. Experiment A in Appendix B is designed to measure these quantities on a heterogeneous population before any claim is made about larger deployments.

Chapter 6

The Idea Router and Computation Without Execution

Not every participating computer needs to perform intensive calculations. Some machines may contribute by routing tasks, indexing capabilities, maintaining provenance records, forwarding experimental observations, or connecting computational problems with suitable resources. An old computer with minimal storage and processing power can still perform useful coordination. This chapter makes the role precise, shows what it costs, and sets out the limits of what a router can legitimately claim to do.

6.1 Calculating and Enabling

The chapter develops the distinction between calculating an answer and enabling a calculation to occur elsewhere. A router does not execute the task. It holds a description of the task, a pointer to where the inputs can be found, and enough metadata to match the task to a capable node. The term *idea router* names this architectural role and carries no claim that messages possess autonomous intelligence. The router matches, forwards, and records. It does not understand, and its usefulness does not depend on any pretense that it does.

6.2 Descriptions, References, and Discovery

A task description must be small enough to move cheaply and specific enough to be matched. Content-addressed references identify data by a cryptographic hash of its contents [7], so that anyone who obtains the data can verify that it is the data referred to, and identical data is recognized as identical wherever it is stored. Discovery of which nodes hold a given piece of data or offer a given capability can be organized as a distributed hash table, in which a lookup takes a number of messages that grows only logarithmically with the number of nodes [30].

The cost of coordination deserves attention because it is the quantity most often neglected.

Coordination consumes resources, and a network whose coordination cost grows faster than the work it coordinates has defeated its own purpose. Let n be the number of nodes and suppose total work grows in proportion to n. Table 6.1 compares three ways of organizing discovery, with the overhead of coordination measured per unit of work.

Organization	State per node	Lookup messages	Overhead per unit
Full mesh	$\Theta(n)$	$O(1)$	$\Theta(n)$
Distributed hash table	$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(\log n)$
Central index	$O(1)$ at nodes	$O(1)$	$O(1)$

Table 6.1: Asymptotic coordination cost of three organizations of discovery.

The full mesh fails by the criterion above, since each node must track every other, so that total state grows as n^2. The central index has the lowest overhead and the most serious defect, namely that it concentrates authority over discovery in one place, which conflicts with the local authority of Chapter 17 and creates a single point of failure and control. The distributed hash table accepts a logarithmic growth in overhead to avoid both defects, and it is the reference design. The asymptotics hide constants and the behavior of real, churning populations, so the conclusion is a design preference and not a measured result.

6.3 Routing Reliability

Routers are themselves intermittent, and a message may traverse several of them. If a path has h hops and each hop delivers with probability p, independently, the path delivers with probability p^h, which falls quickly as h grows. Redundant paths restore reliability.

Proposition 6.1. *Suppose a message is sent along k link-disjoint paths, each of h hops, with each hop delivering independently with probability p. The probability that at least one copy arrives is*

$$1 - \left(1 - p^h\right)^k.$$

Proof. A single path delivers with probability p^h, since all h hops must succeed. The k paths are disjoint and therefore independent, so all fail with probability $(1 - p^h)^k$, and at least one succeeds with the complementary probability. □

With $p = 0.95$ and $h = 6$ a single path delivers with probability about 0.74, whereas three disjoint paths deliver with probability about 0.98. The cost is a threefold multiplication of traffic, which belongs in the overhead accounting of the preceding section. The independence assumption is the weak point. Paths that share infrastructure, a common network provider, or a common power outage fail together, and then the formula overstates the benefit of redundancy.

6.4 Matching and Its Limits

A router may propose that two nodes, or two results, be connected: a task with a node able to run it, an anomalous measurement with a node that specializes in detecting anomalies, or a failed experiment with a later attempt at the same problem. The matching criteria must be explicit. When they are declared tags, signatures of data types, or numerical thresholds, the router's proposals can be audited. When they rely on learned similarity measures, the router will produce connections that look plausible and have no validated basis, and a network that acts on such connections without validation fills its record with spurious links. The rule is that a proposed connection is a hypothesis. Any value it is credited with must come from a validation procedure carried out by parties other than the router, as in the separation of recording and validating described below.

6.5 Preserving Negative Results

The chapter also considers how a routing network could preserve unsuccessful experiments and prevent repeated rediscovery of known failures. A record of what has been tried and failed is a reusable product of computation, and it is one that a metric based on completed results alone would discard. Content addressing makes the record cheap to use. If the description of a task is hashed, a node that is about to start the task can look up whether the same task has already been attempted and what happened. If a fraction h of newly submitted tasks have already been attempted and the stored outcome can be trusted, the network avoids the cost of h of its work, at the price of storing the records and of verifying them. The benefit depends on the hit rate, and it is largest in the settings where the space of experiments is large and attempts are expensive, which is where physical experiments of Part III fall.

6.6 Recording and Authority

A routing node can contribute without computing, and an example shows how. Suppose a thermal experiment in one household produces an anomalous pressure measurement. A low-powered machine elsewhere receives only a compact metadata record describing the observation and forwards it to a node that specializes in statistical anomaly detection, which may then request further measurements. The intermediary performs almost no numerical work yet organizes the investigation.

The design therefore keeps append-only records of observations, task assignments, failures, and successful continuations, and it separates two acts that are easily confused: recording an observation and holding the authority to accept it as a validated result. The distinction matters most in a heterogeneous network in which many nodes are unreliable, intermittent, or potentially compromised. A node may be trusted to preserve and transmit a record

without being trusted to certify what the record shows, and the validation procedure of Chapter 16 stands apart from the routing layer for that reason.

6.7 Identity and Attack

Routing in an open network is exposed to attacks that exploit the cheapness of identities. In a peer-to-peer system without a trusted authority to certify identities, a single party can present many identities and so control a large share of the routing structure [14]. An attacker who controls the routers around a victim can isolate it from the rest of the network or filter what it sees. The remedy anticipated in Chapter 16, tying each node to a physical host, raises the cost of identities without eliminating the problem, since hosts can be acquired. The routing layer should be designed on the assumption that some fraction of its nodes are hostile, with redundant paths, verification of content by hash, and no reliance on any single router for correctness.

6.8 Conditions Under Which the Proposal Fails

The routing role fails to justify itself if coordination overhead grows faster than the work coordinated, if redundant paths are not independent enough to deliver the reliability they are paid for, if the matching proposals cannot be validated at acceptable cost, or if the routing layer cannot be made robust against a plausible fraction of hostile nodes. The role is also unnecessary if tasks are so few and so large that discovery is not a bottleneck. Its value lies in settings of many small tasks and many small nodes, which is the setting that Chapter 5 describes.

Chapter 7

Thermally Integrated Digital Computation

The second form places conventional computing hardware within purpose-built heating systems. A GPU or other processor performs useful digital calculations while its electrical consumption contributes heat to the building. The idea is simple to state and easy to overstate, so this chapter is largely an exercise in accounting: what the arrangement saves, what it does not, and under what conditions it earns its place.

7.1 Thermodynamic Accounting

Electricity consumed by computing hardware leaves it as heat. A negligible fraction leaves as signal energy carried by network links, and the mechanical work of fans ends as heat in the airstream. For a module integrated into a heating system, the heat delivered to the conditioned space is therefore approximately the electrical input,

$$Q_{\text{heat}} = f\, E_{\text{electric}} \approx E_{\text{electric}},$$

where f is the fraction of the dissipated energy that actually reaches the space to be heated, close to one for a properly integrated module and much lower for hardware that happens to sit in an unconditioned basement. The computing system does not create energy. Landauer's analysis of the heat cost of irreversible computation [26] sets a floor far below anything practical hardware approaches, so for engineering purposes essentially all consumed electricity appears as heat.

The consequence is that a computing heater and a resistance heater of the same rated power deliver the same heat from the same electricity, and the computing heater also produces calculations. This is not free energy and not a violation of any thermodynamic bound. It is the observation that the electricity was going to be dissipated as heat in any case, and that it can pass through a useful computation on the way.

7.2 The Heat-Pump Comparison

The arrangement must be distinguished from the heat pump, which moves heat from a cold reservoir to a warm one and so can deliver several units of heat per unit of electricity. Let $\gamma \geq 1$ denote the coefficient of performance of the baseline heating appliance, defined as the heat delivered divided by the electricity consumed, with $\gamma = 1$ for resistance heating. Let p be the price of electricity per kilowatt-hour, let v be the value of the computation produced per kilowatt-hour consumed by the computing heater, and let k be the additional non-energy cost per kilowatt-hour of heat delivered, covering annualized capital, maintenance, networking, and hardware wear.

Proposition 7.1. *Assume $f = 1$. Relative to a baseline heating appliance of coefficient of performance γ delivering the same heat, a computing heater is economically justified only if*

$$v \;\geq\; p\Big(1 - \frac{1}{\gamma}\Big) + k.$$

For a resistance baseline ($\gamma = 1$) the condition reduces to $v \geq k$.

Proof. To deliver one kilowatt-hour of heat the baseline consumes $1/\gamma$ kilowatt-hours of electricity and the computing heater consumes one. The extra electricity costs $p(1 - 1/\gamma)$, to which the additional cost k is added. The computing heater produces value v per kilowatt-hour consumed, which is value v per kilowatt-hour of heat delivered. The arrangement pays only if that value covers the extra cost. □

The proposition shows why the comparison depends so strongly on the baseline. Against resistance heating the computation need only cover its own extra costs. Against a heat pump with a coefficient of performance near three, the computing heater uses roughly three times the electricity for the same heat, and the computation must be valuable enough to pay for the difference. The coefficient of performance of an air-source heat pump also falls as outdoor temperature drops, which narrows the gap in the coldest hours, but that dependence has to be measured on the equipment in question and not assumed from nameplate figures.

7.3 An Admissibility Condition

Let $D_i(t)$ denote the heating demand of household i, expressed as the electrical power that direct resistance heating would need to meet it. Let $P_{\max}$ be the maximum power draw of the computing hardware, and let $A_i(t) \in \{0, 1\}$ indicate whether the queue holds useful, interruptible work at time t. The computing power is

$$P_i(t) = A_i(t)\, \min\{P_{\max},\, D_i(t)\},$$

so that $0 \le P_i(t) \le D_i(t)$ always holds, and a conventional resistance element supplies the remainder,

$$P_i^r(t) = D_i(t) - P_i(t).$$

The network seeks to maximize $\int C_i(t)\,dt$, with $C_i(t)$ the useful throughput, subject to the household's thermal requirements, hardware constraints, and the availability of legitimate work.

Proposition 7.2. *Under this rule the household's electricity consumption at every instant equals $D_i(t)$, the consumption of a resistance heater meeting the same demand, regardless of how much work the queue contains. Computation is a pure addition to the heating service, and the occupant's thermal requirement is met whenever the appliance can deliver $D_i(t)$.*

Proof. Total electrical power is $P_i(t) + P_i^r(t) = D_i(t)$ by construction. Heat delivered equals electricity consumed under the accounting above, so the demand is met at each instant at which the combined appliance can supply it. □

The proposition is the formal content of the governing principle that the building's heating requirement determines the permissible computation, and not the reverse. It also disposes of the objection that the hardware would be run to produce heat nobody needs. Under the rule it cannot be, because the power it draws is capped by demand and the remainder is supplied by the cheaper element. Expressed through the counterfactual test of Chapter 4, every kilowatt-hour consumed would have been consumed anyway.

7.4 Storage and Timing

Thermal storage, in the form of a hot-water tank or a thermal mass, decouples the time at which heat is produced from the time at which it is needed, and it allows computation to run when work is plentiful or electricity is cheap. It also threatens the principle, because stored heat that is never used is a pure waste. The rule that preserves the principle is that computation may run ahead of demand only into storage that the household will draw on within a bounded horizon. Heat must never be discarded in order to keep the hardware busy, and a forecast of demand, not the availability of work, sets the limit.

Interruption has a cost of its own. A workload that is stopped when the thermostat is satisfied must be able to checkpoint cheaply and resume elsewhere, which restricts the work to tasks that are divisible, restartable, and tolerant of delay. The suitable workloads are batch inference, scientific simulation, and rendering. Latency-sensitive services and long jobs that cannot be checkpointed are poorly suited to a duty cycle dictated by a thermostat.

7.5 Attributable Electricity per Unit of Computation

The comparison with a data center can now be made at the level of attributable electricity. Let e denote the electricity consumed by the processing hardware per unit of computation.

A data center spends $e \cdot \mathrm{PUE}$ per unit, where the power usage effectiveness is the ratio of total facility energy to the energy delivered to the computing equipment, and is at least one. A computing heater, in hours when demand covers the hardware's draw, spends an attributable amount

$$e\Big(1 - \frac{1}{\gamma}\Big)$$

per unit, since the fraction $1/\gamma$ of that electricity would have been consumed for heating regardless. With a resistance baseline the attributable electricity is zero, and with a heat pump it is a fraction of e. The expression holds only in hours when demand covers the hardware's draw. In other hours the hardware cannot run without wasting heat, which is why the admissibility condition above bounds it.

This advantage in operating energy has to be set against a disadvantage in embodied cost. The utilization of the hardware over its life is limited by the heating season and by the duty cycle the thermostat allows. The embodied energy and cost of manufacture are then spread over fewer computations than in a data center that runs its processors close to continuously. The same quantity that makes the arrangement attractive on operating energy makes it unattractive on embodied energy, and a fair assessment of the two requires both, a point taken up with the full accounting of Chapter 13.

7.6 Hardware Lifetime and Modularity

A furnace or a heating module is designed to last decades, and computing hardware becomes obsolete in a few years. A design that binds the two together will either retire a working heater along with each processor generation or leave the household with obsolete processors in a functioning appliance. The more sensible arrangement separates them: a durable heating body with a standard mounting and a replaceable computing cartridge, so that the heater persists while its computing hardware is exchanged on its own schedule. The residual value of retired cartridges then becomes a question for the reuse economics of Part IV.

7.7 Safety, Authority, and Deployment

The processors would have to be integrated into a certified heating appliance with proper electrical protection, airflow, temperature limits, and independent thermal shutdowns that the computational scheduler cannot override. Substituting processors for the burner of an existing gas or oil furnace would be neither safe nor practical. An engineered electric duct-heating module or a hydronic heat exchanger is the more plausible starting point. The two present different problems. An air-cooled duct module is simple but couples chip temperature to the airstream, whereas a liquid-cooled hydronic module can deliver heat to a radiant system or storage tank at temperatures that must be reconciled with the temperatures at which processors run reliably, a question to be measured and not assumed.

The occupant retains authority over the thermostat, over participation, over privacy,

and over the hardware, in line with Chapter 17. The network's workloads run in isolation from the household's own network, and no calculation, however valuable, may command more heat than the occupant requests.

7.8 Conditions Under Which the Proposal Fails

The chapter closes by stating where the argument would not hold, since a proposal that cannot fail has not been stated. It fails when the coefficient of performance of the baseline is high enough that Proposition 7.1 cannot be satisfied at realistic values of v and k. It fails when heating seasons are too short, or the duty cycle too low, for embodied costs to be recovered. It fails when the value of the work that can be interrupted and resumed is too small to cover coordination and networking. And it fails when the useful work cannot be verified independently of those who profit from it, which brings the argument back to the governance problem of Part V. Experiment B in Appendix B is designed to measure the first three of these on real hardware over a heating season.

Chapter 8

Physical Analog Computation

The third form uses actual physical processes to perform calculations or experimental evaluations. Instead of digitally approximating every stage of fluid evolution, a controlled physical apparatus evolves under the relevant physical laws and produces measurable outputs. The proposal is the least conventional of the three forms and the easiest to misdescribe, since nearly any physical process can be called a computation if the criterion is loose enough. This chapter therefore concentrates on the criterion: what has to be specified before a physical process counts as computing, how its usefulness is measured, and when it is cheaper than the digital alternative.

8.1 Precedents

Analog computation has a long record. Bush's differential analyzer solved ordinary differential equations mechanically in the early 1930s [10], and for decades analog machines were the practical means of simulating systems whose digital treatment was too slow. Physical reservoir computing is a modern descendant. It uses the transient dynamics of a fixed physical system as a high-dimensional nonlinear transformation of its input, and it trains only a simple readout from measured states [28, 39]. A well-known demonstration used the surface of a bucket of water as the reservoir for a pattern recognition task [16]. Experimental system identification, the estimation of a system's behavior from measured inputs and outputs, supplies the complementary tradition [27]. What these have in common is that the physical dynamics do work that a digital machine would otherwise do, and that a measurement stage converts the result into symbols.

8.2 A Formal Description

A physical process can be represented as

$$y(t) = \mathcal{F}_\theta[u(t)],$$

where $u(t)$ represents controlled inputs, θ represents the apparatus configuration, and $y(t)$ represents measurable outputs. Real apparatus is also subject to variation that is not controlled, which is included by writing

$$Y = \mathcal{F}(\theta, u, \xi),$$

where ξ stands for ambient temperature, supply fluctuations, sensor noise, and every other uncontrolled factor. In a reservoir arrangement the measured state vector $x(t)$ is read out through a trained linear map, $\hat{y}(t) = W_{\text{out}}\, x(t)$, and only W_{out} is learned. The reservoir is useful when its dynamics are nonlinear enough to separate inputs that must be told apart, and when its memory of past inputs fades quickly enough that the response is reproducible.

The crucial question is whether the physical evolution can be configured to answer a specified computational or scientific question. That requires a definition of what answering means.

8.3 Computation, Experiment, and Noise

Definition 8.1 (Experimental computation)**.** An *experimental computation* consists of a physical protocol, specified by a configuration θ and controlled inputs u, together with a measurement map and a decision rule, and a question posed in advance with a set of candidate answers, such that the measured outputs discriminate among those answers with a stated reliability across the uncontrolled variation ξ.

The definition separates three cases that are easily confused. A genuine analog computation has all the elements above: the apparatus is configured, inputs are controlled, and the output answers a question that was posed before the run. A scale-model channel built to determine how pressure loss varies with the geometry of a passage is a computation in this sense. An experiment that merely produces data lacks the question, or acquires it only after the fact, so that the observations are available to be interpreted but were not designed to discriminate among answers. An uncontrolled process whose complexity is mistaken for useful information lacks everything. A turbulent plume observed without a hypothesis is not computing anything, however intricate it looks, and neither is a refrigeration loop that merely runs. The mistake to avoid is treating complexity as if it were computation.

Discrimination can be made quantitative in two ways. For two candidate answers whose predicted outputs differ by $\Delta\mu$, with noise of standard deviation σ across ξ, the discriminability is

$$d' = \frac{\Delta\mu}{\sigma},$$

and a protocol is useful for that question only if d' is large enough for the required reliability. More generally, if H ranges over the hypotheses under test, the information the protocol is

expected to supply about H is the mutual information

$$I(\theta) = I(H; Y \mid \theta, u),$$

which is the quantity maximized over admissible configurations in Chapter 10. Repeating a run n times reduces the noise in the average roughly as $\sigma/\sqrt{n}$ if the repetitions are independent, so information per run saturates while cost grows linearly, and the protocol has a natural stopping point.

8.4 Similarity and Scaling

An apparatus answers a question about another system only if the two are physically similar in the relevant respects. Dimensional analysis [9] expresses this through dimensionless groups. For fluid flow the Reynolds number

$$\mathrm{Re} = \frac{\rho U L}{\mu},$$

with density ρ, characteristic speed U, length L, and dynamic viscosity μ, governs the balance of inertial and viscous effects, and the Mach number U/c, with c the speed of sound, governs compressibility. A small apparatus cannot in general match every group of the full-scale system at once. Reducing L lowers Re unless U rises or the fluid changes, and raising U to compensate changes the Mach number. A module is therefore valid for a particular question only when the groups that matter for that question are matched, and the module's documentation must say which they are. This is the distinction between physical similarity and superficial resemblance, and Chapter 9 applies it to the specific case of heating equipment.

8.5 When Physical Computation Is Cheaper

The case for a physical computer is strongest where digital simulation is most expensive. The cost of resolving all the scales of a turbulent flow numerically grows roughly as the cube of the Reynolds number [32], so that high-Reynolds-number flows quickly become prohibitive. A physical apparatus resolves every scale as a matter of course, since the fluid carries out the evolution whatever the observer can see. Its limitation lies elsewhere, in what can be read out. The bandwidth of the computation is set by the sensors, which sample a few channels of a field that the physics evolves in full. A physical computer is therefore fast in the evolution and narrow in the readout, and the questions it answers well are those whose answers can be expressed through the channels that are measured.

Cost comparison needs a single measure. The thermodynamic computational yield

$$\eta_X = \frac{\text{validated information gained}}{\text{net additional energy consumed}}$$

expresses the information a protocol delivers per unit of energy that it would not otherwise have used. It is not a conventional dimensionless efficiency. It requires a carefully specified measure of information gain, for which $I(\theta)$ above is a candidate, and a consistent boundary for energy accounting. If an experiment yields useful information while supplying heat that would otherwise have required equivalent electrical energy, its incremental energy cost may be small, though not necessarily zero, since pumps, sensors, pressure losses, and changes in heating efficiency still count. A physical protocol beats a numerical one for a given question when its yield exceeds the yield of the simulation, $\eta_X > \eta_{\text{num}}$, with each measured at the same accounting boundary and the same required reliability.

8.6 Calibration, Validation, and Reproducibility

A physical result carries no authority unless it can be reproduced and checked. Sensors drift and apparatus ages, and ambient conditions vary between sites, so each module needs calibration against reference standards, uncertainty estimates attached to every measurement, and periodic validation against a known case. A result accepted into the network's record should state the configuration θ, the controlled inputs u, the measured outputs, the estimated uncertainty, and the calibration state of the sensors, so that an independent party can judge whether the discriminability claimed was achieved. This is the experimental counterpart of the separation between recording and validating introduced in Chapter 6.

8.7 Conditions Under Which the Proposal Fails

Physical analog computation fails when the question cannot be expressed in the channels that can be measured, when the similarity conditions needed to transfer the result to the system of interest cannot be met in a small, safe apparatus, when the noise across sites overwhelms the effect under study so that d' stays small, or when the yield falls below that of simulation at the same reliability. Experiment C in Appendix B is intended to test the last two on a laboratory apparatus before any claim is made about distributed installations.

Part III

Thermodynamic Infrastructure as an Experimental Platform

Chapter 9

Furnaces as Distributed Experimental Instruments

This chapter develops the proposal to use heating systems as experimental platforms, and it serves as the physical test of the monograph's central claim. The architecture of the preceding chapters is worth building only if it produces information that is genuinely valuable and cannot be had more cheaply elsewhere, and not merely by describing ordinary thermodynamic activity as computation. The motivating examples are turbulence, pressure dispersion, mixing, thermal gradients, and flow behavior relevant to engineering. The chapter argues that the proposal is sound in a narrower form than its first statement suggested, and that the narrower form has a clearer target.

9.1 What a Residential System Can Test

The proposal is not to reproduce a complete jet engine inside a residential furnace, and the physics shows why the limits are tighter than that. Flows inside a jet engine involve transonic and supersonic speeds, high pressures and temperatures, and combustion. A residential forced-air system moves air at a few meters per second through ducts of a few tenths of a meter in diameter, at temperatures a little above room temperature. For a duct of hydraulic diameter 0.3 m carrying air at 5 m/s, with kinematic viscosity near $1.5 \times 10^{-5}\,\mathrm{m}^2/\mathrm{s}$, the Reynolds number is of order 10^5, so the flow is fully turbulent, while the Mach number is about 0.015, so compressibility is negligible. The regime is incompressible, turbulent, internal flow.

That regime is not trivial. It is the regime of heat exchangers, ducts, diffusers, mixing sections, and ventilation, and the phenomena that matter there, including pressure loss in passages, turbulent mixing, boundary-layer behavior, heat transfer enhancement by surface structures, and the dispersion of a warm plume, are largely independent of Mach number. A residential module can therefore address questions about components of that kind. It cannot address questions that depend on compressibility, combustion chemistry, or extreme

temperature, and those remain in specialized facilities, in line with the safety envelope described below.

The honest target of the proposal is accordingly not a stand-in for a jet engine but the heating and ventilation equipment itself, studied in place. A furnace network that measures furnaces, ducts, and heat exchangers under real operating conditions is measuring the target system, and not a scaled model of some other one.

9.2 Similarity and Scaling

The chapter examines Reynolds similarity, dimensional analysis [9], boundary conditions, scaling limitations, and uncertainty quantification. For forced convection of a fluid with constant properties through geometrically similar passages, dimensional analysis reduces the dependence of the Nusselt number to two groups,

$$\mathrm{Nu} = \frac{hL}{k} = \phi(\mathrm{Re}, \mathrm{Pr}), \qquad \mathrm{Pr} = \frac{\nu}{\alpha_{\mathrm{th}}},$$

where h is the heat transfer coefficient, k the thermal conductivity, ν the kinematic viscosity, and α_{th} the thermal diffusivity. When the module and the target both use air at similar temperatures, Pr is matched automatically and only the Reynolds number remains to be matched. Buoyancy adds a further group, the Richardson number $\mathrm{Ri} = g\beta\Delta T\, L/U^2$, which compares buoyant to inertial forces and can matter at the low speeds and modest temperature differences of residential ducts.

Matching Re at a reduced scale $s = L_{\mathrm{model}}/L_{\mathrm{full}}$ with the same fluid requires $U_{\mathrm{model}} = U_{\mathrm{full}}/s$. The Mach number then rises by a factor $1/s$. If the full-scale Mach number is 0.015 and compressibility effects are treated as negligible below about 0.3, scale reductions of order twenty are possible before the incompressible approximation fails, though the fan power needed rises steeply with the speed. The accounting shows that Reynolds matching is feasible for modest scale factors within the low-Mach regime and becomes impossible when the target involves high Mach numbers, where the two groups cannot both be matched in a small apparatus. The distinction between physical similarity and superficial resemblance is central. A module is valid for a question only when the groups that govern that question are matched, and the claim of similarity must be argued, not assumed.

9.3 Measurement and Uncertainty

A measurement is useful only with a stated uncertainty. The standard procedure [23] combines the uncertainties of the inputs to a derived quantity $f(x_1, \dots, x_n)$ by the law of propagation,

$$u_c^2(f) = \sum_{i=1}^{n} \Big(\frac{\partial f}{\partial x_i}\Big)^2 u^2(x_i),$$

for uncorrelated inputs, with correlation terms added otherwise. A module reports temperature, differential pressure, and flow rate with calibrated sensors, and every derived quantity, such as a loss coefficient or a heat transfer coefficient, carries the combined uncertainty. Household conditions add uncontrolled variation ξ in the notation of Chapter 8: duct leakage, filter loading, building infiltration, and ambient temperature all change from site to site. These are not measurement errors in the usual sense. They are differences in the system being measured, and they must be estimated and not ignored.

9.4 Site Effects and Identifiability

The distributed design has a statistical weakness that must be dealt with in the design of the network. Suppose the outcome at site i under configuration θ_j is modeled as

$$Y_{ij} = \mu(\theta_j) + b_i + \varepsilon_{ij},$$

where $\mu(\theta_j)$ is the effect of the configuration, b_i is a site effect with variance σ_b^2, and ε_{ij} is measurement noise with variance σ_ε^2. The evolutionary search of Chapter 10 assigns a different configuration to each module, and that choice has a consequence.

Proposition 9.1. *If each configuration is tested at exactly one site and no common reference configuration is run, the configuration effects $\mu(\theta_j)$ and the site effects b_i are not separately identifiable.*

Proof. With one configuration per site, the observation at site i depends only on the sum $\mu(\theta_{j(i)}) + b_i$, however many repeated runs are made. Replacing $\mu(\theta_{j(i)})$ by $\mu(\theta_{j(i)}) + \delta_i$ and b_i by $b_i - \delta_i$ for arbitrary δ_i leaves every observation distribution unchanged, so the parameters cannot be recovered separately. □

The remedy is to connect the design. If every module periodically runs a common reference configuration θ_{ref}, the site effect is estimated from the reference runs and subtracted from the others. It is also enough for configurations to be shared across sites in a pattern that links them into a connected graph, so that differences between sites can be inferred through the shared configurations. With reference runs the variance of the estimated effect of a configuration is reduced to the noise of the run plus the uncertainty in the site correction, whereas without them the effect of the site simply adds to the error of the comparison. The cost is that a fraction of each module's time is spent on the reference configuration, which lowers the rate of exploration and has to be counted as part of the price of the information.

9.5 When the Network Beats the Laboratory

The comparison that decides the question is cost per validated measurement against a purpose-built laboratory rig. The network has real advantages. It runs many configurations

in parallel, it samples genuine field conditions, it produces long time series across seasons, and the heat it generates is useful to the household and not wasted. It has equally real disadvantages: the conditions are less controlled, sensors are cheaper and less accurate, the safety envelope limits the range of experiments, and communication adds overhead. In the notation of Chapter 8, the network wins where its yield η_X exceeds that of the laboratory for a question that can be answered at its accuracy.

The class of questions in which the network is strongest is field validation: whether a heat exchanger or duct design performs as predicted in actual buildings, and how real installations differ from the idealized conditions of the test rig. These questions have the target system as the subject, so site variation is signal and not only noise, and no laboratory can substitute for installed equipment. The questions in which the network is weakest are those that need tight control of boundary conditions or extreme operating points.

9.6 Safety and Certification

A residential system operates within its ordinary certified thermal and pressure envelope. Extreme-temperature or high-pressure experiments remain confined to specialized facilities. Even passive modules have effects that must be assessed. A set of vanes or surface structures in an air passage raises the pressure drop, which can reduce airflow through the heat exchanger and interfere with the appliance’s temperature limits and safety switches. A module must therefore keep the static pressure and the temperatures within the manufacturer’s rated limits, with margin, and must not provide any access to the combustion side of a fuel-fired appliance. Sensor ports must not create leaks. Modifying a certified appliance can also void its listing, so the module is best installed as a certified accessory or fitted during manufacture. The scheduler cannot override the appliance’s safety controller, as stated in Chapter 17.

9.7 A Staged Test and Its Failure Conditions

The physical test of the architecture proceeds in stages, with pre-registered hypotheses and a commitment to report null results. A laboratory thermal-flow apparatus comes first, corresponding to Experiment C. A small number of certified modules in homes follows, each running the reference configuration, to estimate the site variance σ_b^2 directly. Scaling comes only if that estimate is small enough relative to the configuration effects that matter.

The proposal fails if the site variance dwarfs the effects of configuration, so that the discriminability of Chapter 8 stays low. It fails if the module changes the behavior of the appliance it is installed in, or if the questions the network can answer at its accuracy are not ones that anyone needs answered. It fails, finally, if the yield of the network falls below that of simple laboratory testing, in which case what remains is a heating system with sensors attached, which is a modest result and an honest one.

Chapter 10

Evolutionary Flow Geometry

This chapter develops the proposal for individually configured release boards. Each participating experimental module possesses a parameterized geometry θ. The network observes how different configurations influence flow, mixing, pressure losses, and heat transfer, and it uses what it learns to choose the configurations to test next. The chapter treats the search as an experimental design problem under three constraints that dominate it: each evaluation is a slow physical event, each is noisy and site dependent, and every configuration tested must be safe. Within those constraints it asks when physical evolutionary search is the right instrument and when conventional numerical optimization is better.

10.1 Geometry and the Admissible Set

A module's configuration is a vector $\theta = (a_1, \dots, a_n)$ describing features such as passage geometry, vane angles and spacing, surface structures, and flow restrictions, with each parameter bounded. The set of configurations the network may test is

$$\mathcal{A} = \{\theta : \; g_j(\theta) \le 0, \; j = 1, \dots, J\},$$

where the constraints g_j express limits on pressure drop, temperature, and mechanical integrity. The set $\mathcal{A}$ is fixed by certification and by the manufacturer's limits and not by the optimizer, as in the safety envelope of Chapter 17. Each configuration is a physical object that must be fabricated and installed, so exploration has a cost per configuration that numerical search does not. This favors a library of pre-fabricated, interchangeable inserts over per-experiment fabrication, and it makes the number of distinct configurations a scarce resource that the design should spend deliberately.

10.2 Surrogates and the Information Objective

A surrogate model predicts the outcome of untested configurations from the record of tested ones, and an evolutionary search proposes configurations for subsequent testing. A Gaussian

process is a natural surrogate here [33], since it supplies not only a prediction but also a predictive variance $s^2(\theta)$ at each untested point. For a single measurement $y = f(\theta) + \varepsilon$ with Gaussian noise of variance σ^2, the information that the measurement supplies about $f(\theta)$ is

$$I(f(\theta); y) = \tfrac{1}{2} \ln\Big(1 + \frac{s^2(\theta)}{\sigma^2}\Big),$$

measured in nats. The expression shows how the objective of Chapter 8 behaves. Information is largest where the surrogate is most uncertain and where the measurement is least noisy, and it falls to zero where the surrogate is already certain. The optimization objective must be explicitly constrained:

$$\max_{\theta \in \mathcal{A}} I(\theta),$$

where $I(\theta)$ represents expected scientific information gain and $\mathcal{A}$ is the independently established set of safe, permitted configurations. The system does not optimize pressure, turbulence, or temperature without bounds, and an objective that mixes information gain with a performance target must state the exchange rate between them explicitly, since the two pull in different directions: information gain favors uncertain regions, and performance favors regions already known to be good.

Population-based methods such as evolution strategies with covariance adaptation [19] are the usual tool for proposing candidates when the objective is irregular, and they need many evaluations. The surrogate reduces the number of physical evaluations by screening candidates before any is built. The point of the combination is to spend scarce physical tests where they are most informative.

10.3 The Evaluation Budget

Each physical evaluation requires the flow and temperature to settle and then to be measured, which takes minutes to hours. If a module completes one evaluation every τ hours and N modules operate in parallel for T hours, the network performs about NT/τ evaluations, less the fraction spent on reference runs. For $\tau = 2$ hours and a thousand modules this is on the order of ten thousand evaluations per day. That is a respectable number for a physical experiment and a small one by the standards of numerical search, where a single workstation can evaluate far more candidate designs in a day if a trusted simulation exists. The network's value cannot lie in the volume of evaluations. It lies in the fact that each evaluation is a measurement of the real system.

10.4 Selection Under Noise

Because outcomes are noisy, a selection step that compares two configurations can choose the wrong one. Suppose two configurations differ in true mean outcome by $\Delta > 0$, each is measured m times with independent noise of standard deviation σ, and the one with the

larger sample mean is selected.

Proposition 10.1. *The probability that the truly better configuration is selected is*

$$\Phi\left(\frac{\Delta\sqrt{m}}{\sigma\sqrt{2}}\right),$$

where Φ is the standard normal distribution function. Selection is correct with probability at least $1-\delta$ when

$$m \;\geq\; \frac{2\,z_\delta^2\,\sigma^2}{\Delta^2}, \qquad z_\delta = \Phi^{-1}(1-\delta).$$

Proof. The difference of the two sample means is normally distributed with mean Δ and variance $2\sigma^2/m$. The better configuration is selected when this difference is positive, which has probability $\Phi(\Delta/(\sigma\sqrt{2/m}))$. Setting this at least $1-\delta$ gives $\Delta\sqrt{m}/(\sigma\sqrt{2}) \geq z_\delta$, which rearranges to the stated bound. □

The bound shows that distinguishing small improvements is expensive: halving Δ quadruples the number of replications needed. Evolutionary search that ignores the noise will drift on spurious differences, so the search must either replicate or accept a stated rate of selection error.

The proposition assumes that both configurations were measured under the same conditions, which in a distributed network they are not. If they are measured at different sites, the site effect enters.

Proposition 10.2. *Using the model of Chapter 9, let each configuration be measured m times at its own site, with site effects of variance σ_b^2 and measurement noise of variance σ^2. The difference between the two sample means then has variance $2\sigma_b^2 + 2\sigma^2/m$. If instead both configurations are measured at the same site, in alternation so that the site effect is common, the variance of the difference is $2\sigma^2/m$.*

Proof. A sample mean at one site has variance $\sigma_b^2 + \sigma^2/m$, and two sites are independent, so the variances add. At a single site the common term b_i cancels in the difference, leaving the noise terms of the two means. □

The comparison across sites has a floor of $2\sigma_b^2$ that no number of replications removes. When the site variance is large relative to the effects of interest, replication does not help and the design must change, either by comparing configurations within a site by alternation, or by running the reference configuration at every site as established in Chapter 9. The alternating design assumes that conditions drift slowly relative to the alternation, which has to be checked.

10.5 Safe Exploration

The constraints in $\mathcal{A}$ are not always known for a new configuration before it is tested. The pressure drop of an untried geometry, for instance, can only be estimated. Exploration must

therefore proceed from a certified starting configuration and expand cautiously. A method with a known structure is to treat the constraint values as unknown functions modeled by the surrogate and to test only configurations whose predicted constraint value, plus a multiple of its predictive standard deviation, lies within the limit [38]. The set of configurations deemed safe grows as data accumulates. The guarantee holds only if the model's assumptions are correct, and a model that is wrong about the constraint can recommend an unsafe test. For that reason the independent safety controller of the appliance remains the last line of defense, and no exploration rule is permitted to override it.

10.6 Comparison with Numerical Optimization

The chapter compares evolutionary physical experimentation with conventional numerical optimization, computational fluid dynamics, and automated laboratory systems, with attention to cost per informative experiment and to the fidelity of the physical module relative to the target system. The strongest competitor is gradient-based design with adjoint methods, whose cost of computing the gradient of an objective is nearly independent of the number of design parameters [21]. Where a trustworthy simulation of the target exists, adjoint optimization of a high-dimensional shape is very hard to beat. Physical evolutionary search has the advantage in the cases where the simulation is the weak link: complicated surfaces and turbulence for which models are unreliable, effects of installation and aging that simulations omit, and objectives that depend on how real systems behave over seasons. In those cases the physical measurement supplies what the model cannot.

10.7 A Distributed Dataset

Each participating module reports a configuration θ_i, together with controlled operating conditions u_i and a measured outcome $Y_i = \mathcal{F}(\theta_i, u_i, \xi_i)$. Across N modules the network accumulates

$$\mathcal{D} = \{(\theta_i, u_i, Y_i)\}_{i=1}^{N},$$

which a central or federated learning system can use to train surrogate models, detect unexpected behavior, and select subsequent experiments. The advantage is not that physical experiments are faster than numerical simulation. It is that they observe actual physical processes, including effects that numerical models approximate poorly, which makes the network most useful for model validation, system identification, and experimental optimization. The obstacle is standardization. Household heating systems differ enormously and their environments are hard to control, so a scientifically useful network needs calibration, uncertainty estimates, reproducibility procedures, carefully controlled experimental modules, and the connected design that Proposition 9.1 shows to be necessary.

10.8 Conditions Under Which the Proposal Fails

Evolutionary physical search fails when the number of configurations that can be fabricated and tested is too small to explore a useful region, when the noise and site effects are too large for selection to be reliable at affordable replication, when safe exploration cannot be assured without the surrogate being right, or when a trustworthy simulation exists that makes physical testing unnecessary. In any of these cases the network is better used for validation of designs found by other means than for discovery.

Chapter 11

Air Conditioners and Refrigeration as Physical Computers

Cooling systems provide a complementary application. An air conditioner is a heat-transfer machine that moves thermal energy between reservoirs. Its refrigerant cycle involves compression, expansion, phase transitions, pressure differences, and heat exchange. These processes could, in principle, support certain forms of physical measurement, system identification, or analog computation. This chapter asks which of those descriptions survives scrutiny. The conclusion is that the loop is a poor analog computer and a good source of information about itself, and that the proposal is better described as instrumenting air conditioners than as converting them into calculators.

11.1 The Cycle as a Physical System

A vapor-compression cooling system moves heat from a cold space to a warm one by compressing a refrigerant, rejecting heat in a condenser, expanding the refrigerant through a throttling device, and absorbing heat in an evaporator. Its coefficient of performance for cooling, the heat removed divided by the work supplied to the compressor, is bounded by the Carnot value,

$$\mathrm{COP} \;\le\; \frac{T_c}{T_h - T_c},$$

with the evaporator and condenser temperatures T_c and T_h expressed in kelvin. For reservoirs at 280 K and 310 K the bound is about 9.3, and real units operate well below it. The cycle has measurable state: suction and discharge pressures, evaporator and condenser temperatures, the degrees of superheat and subcooling, compressor power, and airflow. These signals are produced during ordinary operation and constitute the output Y in the notation of Chapter 8, with the thermostat's demand and the ambient conditions as controlled and uncontrolled inputs.

11.2 What Question Can the Loop Answer

The chapter distinguishes exploiting the physical evolution of a refrigeration system from merely attaching a GPU to an air conditioner. An air conditioner cannot be assumed to provide useful analog computation simply because it contains complex thermodynamics. The experiment must define inputs, measurable outputs, reproducibility conditions, and an independently meaningful inference task, in the sense of Definition 8.1. Three kinds of task can be considered.

The first is inference about the unit itself. Signals from the loop can be used to estimate whether the refrigerant charge is correct, whether the heat exchangers are fouled, and whether the compressor has degraded. This is fault detection and diagnosis, a well-developed field for building systems [24]. It is not analog computation of an external problem. It is inference from measurement, and its value is that it answers a question that owners and operators actually have.

The second is measurement of something external. A secondary circuit thermally coupled to the loop could be used as a calorimeter, for example to characterize the heat capacity or conductivity of a fluid or material placed in it. The loop supplies a controlled thermal environment and the measurement is of the sample.

The third is analog computation proper, in which the loop's dynamics transform a time-varying input into an output that a readout can use. Here the loop is poor. A reservoir with a dominant time constant τ_s responds to input variations only up to a bandwidth of about $1/(2\pi\tau_s)$. For a thermal time constant of five minutes, this is roughly 5×10^{-4} hertz, so the loop can distinguish about one independent input value every ten to twenty minutes. That is a throughput of under a hundred values per day, many orders of magnitude below any digital machine, and no argument about energy cost can make up for it. The loop is of interest for what it measures and not for how fast it computes.

11.3 Fault Detection Across a Fleet

The distinctive contribution of a network is the ability to compare units. Let COP_i be the measured performance of unit i and let $\widehat{\mathrm{COP}}_i$ be the performance predicted for it from the ambient temperature, load, and the fleet's accumulated data. The residual

$$r_i = \mathrm{COP}_i - \widehat{\mathrm{COP}}_i$$

is standardized by the spread σ_r of residuals across comparable units, and unit i is flagged when $|r_i|/\sigma_r > z$. If residuals are approximately normal, the expected number of false flags among N healthy units in one assessment is

$$N \cdot 2\big(1 - \Phi(z)\big),$$

which for $z = 3$ is about $0.0027N$. A fleet of ten thousand units would produce about twenty-seven false flags per assessment, so the threshold, the frequency of assessment, and the cost of a false alarm have to be set together. The comparison across units has an advantage over a single unit's history: a unit that has always performed poorly shows no deterioration but stands out against its peers.

11.4 Non-Invasive Instrumentation

The loop contains refrigerant under pressure, and regulations govern who may open it and how. Any experimental module must preserve cooling performance, refrigerant containment, electrical safety, and the manufacturer's operating limits. The practical approach is to instrument the outside of the circuit and not the inside. Clamp-on temperature sensors on the refrigerant lines, a current transducer on the compressor supply, and air-side temperature and flow measurements require no access to the refrigerant. Pressure can be inferred from the temperature of the two-phase regions, where the refrigerant is at its saturation temperature, using property data for the refrigerant in use [6]. The inference carries its own uncertainty, which has to be propagated as in Chapter 9, and it is valid only where the fluid is in fact a two-phase mixture. A secondary circuit coupled through an external heat exchanger leaves the refrigerant circuit untouched and keeps the experiment on the owner's side of the safety boundary.

11.5 Admissibility and the Thermostat

The principle that governs heating applies equally to cooling. A cooling system operates when the thermostat demands cooling, and the experiment may select among settings the manufacturer permits, such as airflow within its rated range, but it may not command operation that the occupant's comfort does not require. Extra start and stop cycles are a particular hazard, since short cycling shortens the life of compressors, so the experiment must respect the manufacturer's minimum on and off times. In the notation of Chapter 7, the cooling demand plays the role of $D_i(t)$, and the experiment is a pure addition to the cooling service only if it never causes the unit to run when it would otherwise be off.

11.6 Seasonal Complementarity

The chapter introduces a geographically distributed seasonal architecture. Heating-related computational opportunities may increase during cold periods, while cooling-related experimental opportunities may increase during warm periods. Heating and cooling demand are commonly measured by degree days, with

$$\mathrm{HDD} = \sum_{d} \max(0,\, T_{\mathrm{base}} - T_d), \qquad \mathrm{CDD} = \sum_{d} \max(0,\, T_d - T_{\mathrm{base}}),$$

summed over days with mean temperature T_d relative to a base temperature. A network that spans climates, or both hemispheres, has opportunities available through more of the year than one confined to a single region, because the seasons of the two hemispheres are opposed. The complementarity is limited by where people live, since much of the world’s population and equipment is in the northern hemisphere, so the gain in smoothing the annual cycle should be estimated from the actual distribution of participants and not assumed. Reversible heat pumps, which both heat and cool, bring both opportunities into a single appliance.

11.7 Conditions Under Which the Proposal Fails

The cooling proposal fails if the questions the loop can answer, principally the condition of the unit, are not worth the cost of the instrumentation, if the inferences from non-invasive measurements are too uncertain to separate faulty units from healthy ones, if false alarms overwhelm genuine findings, or if the experiment cannot be carried out without altering the duty cycle or the safety of the equipment. It is also a failure of the original ambition if the loop’s slow dynamics are the only basis on which it was proposed as a computer. In that case the remaining result is real but modest: a fleet of air conditioners that report on their own condition.

Chapter 12

Entropy as a Service

The chapter explores the possibility of using physical fluctuations and thermal processes as sources of randomness. Randomness is the one product of the network that does not require a question to be posed in advance, which makes it attractive as a by-product of processes that are already running. It is also the product about which careless claims are easiest to make, because a signal can look random without being unpredictable to anyone who matters. The chapter develops what has to be established before a physical process may be offered as a source, what a distributed service of such sources can and cannot guarantee, and whether the service is worth providing at all.

12.1 Four Notions of Entropy

Thermodynamic entropy, $S = k_B \ln \Omega$, counts the physical microstates compatible with a macroscopic description. Shannon entropy, $H(X) = -\sum_x p(x) \log p(x)$, measures the average uncertainty of a distribution. Min-entropy, $H_\infty(X) = -\log \max_x p(x)$, measures the difficulty of guessing the most likely outcome. Cryptographic unpredictability concerns the probability that an adversary with specified knowledge and capabilities guesses a value, which is governed by the conditional min-entropy $H_\infty(X \mid E)$ given the adversary's view E. The quantities are ordered, $H_\infty(X) \le H(X) \le \log |\mathrm{supp}(X)|$, and the gaps can be large. A source that emits one value with probability one half and a million others with equal small probability has high Shannon entropy and a min-entropy of a single bit. A hot, turbulent process may possess enormous thermodynamic entropy and still deliver samples that an observer with a sensor in the same room can largely predict. Only the last notion bears on security.

12.2 Characterizing a Physical Source

A raw measurement from a physical process is generally biased and correlated, and its distribution is not known in advance. Thermal noise in a resistor, shot noise, and timing

jitter in oscillators are the traditional sources, and turbulent flow adds a chaotic process in which tiny perturbations are amplified, so that the outcome of a given run is in principle highly sensitive to microscopic conditions. Sensitivity to initial conditions is not unpredictability to an adversary, however, who may observe much of the macroscopic state. A source model must therefore state what is assumed, namely which variables are controlled, which are public, and which are private, and then estimate a lower bound on the min-entropy per sample conditional on what the adversary can observe.

Practical procedure is standardized. The NIST recommendation on entropy sources [40] specifies how to estimate min-entropy from samples, how to condition the raw output, and which health tests to run continuously. These include a repetition count test, which detects a source stuck at one value, and an adaptive proportion test, which detects a collapse in the frequency of a value. A source that fails a test must stop contributing until it is reassessed. Estimators bound the entropy from below on the data they see and cannot certify behavior under conditions they have not seen, so the estimate must be conservative and must be repeated when conditions change.

12.3 Extraction and Its Limits

Raw samples with min-entropy rate h per sample are converted into nearly uniform bits by an extractor. The central result is the leftover hash lemma [20]: if a source has min-entropy at least k bits and a hash function is drawn from a two-universal family, the hash output of m bits is within statistical distance

$$\varepsilon \leq \tfrac{1}{2}\sqrt{2^{\,m-k}}$$

of uniform, in the sense that the pair consisting of the hash function and its output is within that distance of the same hash function paired with a uniform string. The entropy bound k must hold conditional on everything the adversary observes, since the guarantee is lost if the adversary knows a function of the source. The practical content is a budget. One may extract at most $m = k - 2\log(1/2\varepsilon)$ bits of nearly uniform output, so the output length is the entropy collected minus a safety margin, and a service that emits more bits than it has collected is manufacturing apparent randomness. The seed that selects the hash function may be public, but it must be independent of the source.

There is also an impossibility that explains why combining sources is not trivial. No deterministic function of a single weak source of the semi-random kind extracts even one nearly unbiased bit for every such source [34]. Extraction from one source therefore needs a seed, and without a seed it needs several sources that are independent of one another. The assumption of independence is the weakest point of any scheme that pools physical measurements, because sources that share an environment are not independent.

12.4 Combining Many Untrusted Sources

The tempting design is to combine many contributions and to trust the result because no single contributor is trusted. Two propositions show what this does and does not buy.

Proposition 12.1. *Let $X_1, \ldots, X_n$ be independent bits with $\Pr[X_i = 1] = \frac{1}{2} + \epsilon_i$. Then the exclusive-or $Z = X_1 \oplus \cdots \oplus X_n$ satisfies $\left|\Pr[Z = 1] - \frac{1}{2}\right| = 2^{n-1} \prod_{i=1}^{n} |\epsilon_i|$.*

Proof. For a single bit, $\mathbb{E}[(-1)^{X_i}] = \Pr[X_i = 0] - \Pr[X_i = 1] = -2\epsilon_i$. By independence, $\mathbb{E}[(-1)^{Z}] = \prod_i (-2\epsilon_i)$. Since $\mathbb{E}[(-1)^{Z}] = -2(\Pr[Z = 1] - \frac{1}{2})$, the magnitude of the bias of Z is $\frac{1}{2} \cdot 2^n \prod_i |\epsilon_i| = 2^{n-1} \prod_i |\epsilon_i|$. □

The proposition says that independent, individually weak sources become strong rapidly when combined, since each added source multiplies the bias by a factor below one. Its hypothesis, independence, is the whole of the argument, and there are two ways it fails.

Proposition 12.2. *Suppose a single contributor observes the bits of all other contributors before submitting its own, and that its own bit is not constrained by any physical source model. Then it can force the exclusive-or Z to any value it likes.*

Proof. Let S be the exclusive-or of the other contributions and let z be the desired output. The contributor submits $S \oplus z$, so that $Z = S \oplus (S \oplus z) = z$. □

The two assumptions about what the contributor observes and controls are essential. If the contributor sees nothing before committing, or if its output is the reading of a physical source that it cannot choose, the argument does not apply, and the combination retains the entropy of the other contributions. A weaker adversary who sees only some of the contributions can still bias the result, at a cost that depends on how much it sees. One adaptive or compromised participant therefore defeats the combination entirely, no matter how many honest participants there are. A commit-and-reveal protocol removes the ability to choose a value after seeing the others, since each contribution is committed in advance, but it does not remove the ability to abstain, because a final participant who dislikes the outcome can decline to reveal and so bias it. The second failure is correlation, which includes the case of a source that is unpredictable in isolation but predictable when conditioned on information held by another contributor, such as a shared environmental signal. If all the sources are driven by a common influence, such as the same ambient temperature cycle or the same mains frequency, their outputs are not independent, and in the extreme case of identical copies the combination is no better than one source. A service of this kind needs independence tests among contributors, and the independence it can certify is only as good as those tests.

12.5 Is the Service Worth Providing

It must be asked whether entropy is scarce enough to justify a distributed physical service. For most uses it is not. Operating systems collect randomness from many sources, and

current processors provide hardware random number instructions, so the supply of bits is rarely the constraint. The constraint is trust: whether a particular source was honestly built, whether it has been backdoored, whether its output can be audited. A distributed service has a case where independence of manufacture and public auditability are themselves the product, as in public randomness used for lotteries, audits, and protocols that require a value no single party controls. For private cryptographic keys, a local source plus a well-vetted generator, such as the deterministic generators of NIST's companion recommendation [4], serves better than a global network, since sending raw physical samples across a network exposes them. The economic case therefore rests on diversity of provenance more than on volume, and it should be argued in those terms.

12.6 An Admissible Entropy Contribution

The chapter's rule can be stated as a definition parallel to the one in Chapter 18. A contribution is admissible when it comes with a documented source model, a conservative lower bound on min-entropy per sample conditional on the adversary's stated view, passing continuous health tests, a commitment made before the reveal, and a declaration of the environmental influences it shares with other contributors. The service outputs bits in number no greater than the aggregate entropy estimated by that bound, less the margin required by the extraction lemma. Rejecting the assumption that combining many untrusted signals automatically produces trustworthy entropy is not an argument against the service. It is a specification of what the service must check.

12.7 Conditions Under Which the Proposal Fails

The proposal fails if the sources cannot be shown to be unpredictable to an adversary who observes the appliance and its surroundings, if the contributors cannot be shown to be independent, if protocols cannot prevent adaptive or withholding behavior at acceptable cost, or if the demand for diverse-provenance randomness is too small to cover the cost of calibration and testing. The broader question remains whether physical processes already occurring for other reasons can provide valuable stochastic measurements at low incremental cost, and the answer depends on these checks and not on how random the process appears.

Part IV

The Economics of Existing Things

Chapter 13

Sunk Infrastructure and Marginal Computation

This chapter develops the economic case for using already manufactured hardware, and it is the foundation on which the long-horizon argument of Chapter 15 rests. A newly constructed data center and a network of existing household computers have fundamentally different capital-cost structures. The relevant comparison is not simply computational throughput per watt, but the total additional resources required to provide equivalent services. The chapter's first obligation is to avoid a common error: existing hardware is not free hardware. Its manufacturing expenditure may already have occurred, but operating an old computer still consumes electricity, causes wear, occupies space, creates administrative obligations, and may accelerate its replacement.

13.1 Three Kinds of Cost

Any comparison has to say whose costs it counts. Private financial costs are the prices paid and received by the parties, and they depend on tariffs, subsidies, and the compensation offered to owners. Social resource costs are the energy, materials, and labor consumed regardless of who pays. Environmental costs are the emissions and other burdens, including those embodied in manufacture. The three need not agree. A project can be privately profitable while raising total energy consumption, because the energy is cheap or the burden falls on others, and it can be environmentally advantageous while failing to compensate its participants adequately. This chapter states its conditions for each kind of cost separately and does not treat one as a proxy for the others.

13.2 The Comparison Condition

The central proposition is conditional. Let W denote a verified workload, T a horizon, and suppose both arrangements deliver W over T with equivalent reliability, security, and service

requirements. Reuse is preferable when

$$C_{\text{reuse}}(W,T) < C_{\text{dedicated}}(W,T).$$

The equivalence requirement is substantial. Existing machines are unreliable and intermittent, so delivering one verified unit of work requires performing more than one unit of raw work. With the replication of Chapter 16 and the availability of each node taken into account, let $\rho \geq 1$ be the ratio of raw work performed to verified work delivered. The cost of reuse then decomposes as

$$C_{\text{reuse}} = C_{\text{org}} + C_{\text{coord}} + C_{\text{energy}} + C_{\text{wear}} + C_{\text{maint}} + C_{\text{repl}} + C_{\text{comp}},$$

covering organization, coordination, incremental energy, wear, maintenance, accelerated replacement, and compensation to the owners. The cost of the dedicated alternative is

$$C_{\text{dedicated}} = C_{\text{manufacture}} + C_{\text{deploy}} + C_{\text{operation}} + C_{\text{retire}},$$

with the operating term including the overhead of cooling and power delivery. The embodied cost of the existing machine is sunk with respect to a new participation decision, and a full environmental assessment still accounts for hardware lifetime, replacement effects, and disposal.

13.3 Operating Energy and the Strongest Counterexample

The most difficult counterexample is a modern data center that delivers so much more computation per joule that it outperforms the distributed network even after manufacturing is counted. The comparison for operating energy can be written out. Let $\text{PUE} \geq 1$ be the power usage effectiveness of the dedicated facility and e_{new} the electricity per unit of computation on its hardware. For an existing machine that is already switched on, only the energy above its idle draw is attributable to the added work,

$$e_{\text{old}}^{\text{marg}} = \frac{P_{\text{old}}^{\text{busy}} - P_{\text{old}}^{\text{idle}}}{\theta_{\text{old}}},$$

where θ_{old} is its throughput. This matters because computing hardware is rarely energy-proportional, and idle draw is a substantial fraction of peak [5]. The reuse option has an operating-energy advantage only if

$$\rho\, e_{\text{old}}^{\text{marg}} < \text{PUE} \cdot e_{\text{new}}.$$

The inequality shows how much the case depends on the replication factor and on the age of the hardware. A factor ρ of two doubles the old machine's energy per verified unit, and an efficient facility has a PUE near one, so the old hardware must have a marginal energy per

unit that is below half that of new hardware. An older processor that is several times less efficient fails the test on operating energy alone, and many do. The honest conclusion is that reuse of general-purpose old hardware rarely wins on operating energy against a modern facility.

Embodied costs enter the balance on both sides. Let $E^{\text{emb}}_{\text{new}}$ be the emissions embodied in the new hardware and $W^{\text{life}}_{\text{new}}$ the verified work it delivers over its life. For the old machine, participation is charged only the fraction δ by which it shortens the machine's remaining life, applied to the embodied emissions $E^{\text{emb}}_{\text{old}}$ of the machine and spread over the work $W^{\text{life}}_{\text{old}}$ delivered in that life. Reuse is environmentally advantageous per unit of verified work when

$$\rho\,\kappa\,e^{\text{marg}}_{\text{old}} + \frac{\delta\,E^{\text{emb}}_{\text{old}}}{W^{\text{life}}_{\text{old}}} \;<\; \kappa\,\text{PUE}\,e_{\text{new}} + \frac{E^{\text{emb}}_{\text{new}}}{W^{\text{life}}_{\text{new}}},$$

where κ is the carbon intensity of electricity. The embodied term favors reuse, since the existing machine's manufacture is already paid for, and the operating term usually opposes it. A network of general-purpose old computers wins this comparison only where embodied emissions dominate, which is more likely when the grid is clean and less likely when it is not.

Several factors can tip the balance toward reuse, and each should be argued separately and measured. When the electricity would have been consumed for heating regardless, as in Chapter 7, the attributable energy falls sharply and in the resistance-heater case vanishes. When the work is naturally local, or requires low latency or data that should not leave the premises, transmission to a facility has costs that the comparison above omits. And when the value of autonomy, privacy, or resilience is counted, which Chapter 1 argued is not captured by utilization, the preference for local resources has a basis that the cost ledger does not show. Some of these are not priced, and the chapter does not pretend that they are.

13.4 Cost of Participation and Cost of Organizing Participation

The comparison above concerns the marginal cost of using a machine. The network also bears a cost of organizing participation that does not fall when a single household withdraws, including software, protocols, security, verification, and support. Model the cost structures as linear in volume, with reuse costing $F + fW$ and the dedicated alternative costing $K + gW$, where F and K are fixed costs and f and g are costs per unit of verified work.

Proposition 13.1. *If $f \neq g$ the two cost curves cross exactly once, at the volume*

$$W^{\circ} = \frac{K - F}{f - g}.$$

When $f > g$ and $K > F$, reuse is cheaper below W° and the dedicated alternative is cheaper above it. When $f < g$ and $F > K$, the dedicated alternative is cheaper below W° and reuse

is cheaper above it.

Proof. Reuse is cheaper exactly when $F + fW < K + gW$, which is equivalent to $(f - g)W < K - F$. If $f > g$ this reads $W < (K - F)/(f - g)$, and if $f < g$ it reads $W > (K - F)/(f - g) = (F - K)/(g - f)$. The remaining case is that the right-hand side is nonpositive, in which case the inequality holds for no positive volume when $f > g$ and for every positive volume when $f < g$. □

The proposition clarifies the realistic case. If the dedicated facility has high fixed costs and low unit costs, as is typical of data centers (K large, g small), and the network has low fixed costs but higher unit costs because of replication and old hardware ($f > g$), then the network is cheaper at small scale and loses at large scale. That is a precise sense in which the network is a small-scale, low-commitment solution, and it is consistent with the real-options argument of Chapter 15. The network is not expected to displace dedicated infrastructure at large volumes. Whether f can be made smaller than g, so that the network wins above a threshold volume, is an empirical question that Experiment A is designed to inform.

13.5 Rebound and Demand

The comparison above holds the workload fixed, and in practice it is not. Cheaper computation is demanded in greater quantity, and the rebound literature [36] documents cases in which efficiency gains were partly or wholly absorbed by increased use. The classical statement is Jevons's observation about coal [22]. If reuse lowers the marginal price of computation, the workload W can rise, and total energy and total emissions can rise with it even if each unit is cheaper. An assessment of environmental benefit must therefore be made at the level of total consumption, with an estimate of the demand response, and not only per unit of work.

13.6 When Reuse Is Counterproductive

There are circumstances in which lower-performance existing hardware is preferable to new hardware, and circumstances in which it is not. A decade-old processor that performs a given calculation at a tenth of the energy efficiency of current hardware can emit more over a given horizon, through the electricity it consumes, than the manufacture of its replacement would have emitted, and the balance depends on the horizon and on the carbon intensity of the grid. The reasonable policy is to admit hardware into the network on measured marginal energy per unit of work and to exclude machines that fail a stated threshold, and not to treat all existing hardware as a resource. Experiment A in Appendix B supplies the measurements on which that threshold would be set.

13.7 Conditions Under Which the Economic Case Fails

The case fails when the replication factor and the age of the hardware make the operating energy per verified unit exceed that of an efficient facility by more than the embodied savings can offset, when the grid is carbon-intensive enough that operating emissions dominate, when the network's organizing costs are large relative to the volume of work it handles, or when the demand response is large enough that total consumption rises. It also fails when the compensation required to secure participation exceeds the value of the work. In each case the architecture may still be justified on grounds that the cost ledger does not capture, such as autonomy, but those grounds must then be argued as such and not presented as economic efficiency.

Chapter 14

The Geography of Computational Demand

Conventional cloud computing concentrates hardware in locations selected for electricity, networking, taxation, cooling, and institutional convenience. The proposed architecture allows some computational activity to follow geographically distributed needs. Heating demand, cooling demand, available household hardware, communications infrastructure, and experimental opportunities vary by location and season. This chapter asks what that variation implies for where work can be placed, and it finds that the geography both helps and constrains the proposal in ways that are easy to miss.

14.1 Which Work Can Move

Geographical scheduling cannot assume that all computational tasks tolerate relocation. Two physical facts bound what can be moved. The first is delay. Signals in optical fiber travel at roughly two thirds of the speed of light, which is about five microseconds per kilometer, so a one-way trip of ten thousand kilometers takes at least fifty milliseconds and a round trip at least one hundred, before any routing or queuing. Work that needs interactive response from a distant site cannot get it, and no change in architecture removes the limit.

The second is the cost of moving data. Let a task need C units of computation and D bytes of data. Let e_A and e_B be the attributable energy per unit of computation at two sites in the sense of Chapter 7, and let ϵ be the energy needed to transport a byte over the path between them, an empirical quantity that varies widely and has to be measured. Moving the task from A to B saves energy only if

$$(e_A - e_B)\,C \;>\; \epsilon\,D, \qquad \text{that is,} \qquad \frac{C}{D} \;>\; \frac{\epsilon}{e_A - e_B}.$$

The ratio of computation to data decides the matter, in the same way as the efficiency criterion of Chapter 5. Tasks that are small in data and large in computation move easily,

and tasks that are large in data and small in computation do not. A calculation that must read a large archive is better run near the archive.

14.2 Where the Heat Is Needed

The architecture ties computation in heating appliances to heating demand, so the geography of the network is the geography of heating. Demand is concentrated in cold and populated regions, and it varies by season and by hour. Heating degree days, defined in Chapter 11, measure the accumulated demand, and a map of them identifies where thermally integrated computation has the most hours to work with. The consequence is that the network's capacity is concentrated in particular climates, and that the capacity of any one region is not constant but seasonal.

A consequence that is easy to overlook is that capacity is correlated across space. Weather systems cover large areas, and a cold spell raises heating demand over a whole region at once, so the capacity of the network's nodes in that region rises and falls together. Correlation limits how much geographic aggregation can smooth the capacity.

Proposition 14.1. *Let n nodes have capacities $X_1, \dots, X_n$ with common mean μ, common variance σ^2, and common pairwise correlation $\rho \in [0, 1]$, and let $S = \sum_i X_i$. Then*

$$\operatorname{Var}(S) = n\sigma^2\big(1 + (n-1)\rho\big),$$

and the coefficient of variation of the total is

$$\frac{\sqrt{\operatorname{Var}(S)}}{\mathbb{E}[S]} = \frac{\sigma}{\mu}\sqrt{\frac{1 + (n-1)\rho}{n}} \;\longrightarrow\; \frac{\sigma}{\mu}\sqrt{\rho} \quad (n \to \infty).$$

Proof. The variance of a sum is the sum of the variances plus the sum of all covariances. There are n variances equal to σ^2 and $n(n-1)$ ordered pairs with covariance $\rho\sigma^2$, so $\operatorname{Var}(S) = n\sigma^2 + n(n-1)\rho\sigma^2$. Dividing the square root by $\mathbb{E}[S] = n\mu$ gives the coefficient of variation, and $(1 + (n-1)\rho)/n \to \rho$ as $n \to \infty$. □

The coefficient of variation can equivalently be written as $(\sigma/\mu)\sqrt{\rho + (1-\rho)/n}$. The result says that adding more nodes in the same climate stops reducing the relative fluctuation of total capacity once the number of nodes is large, and leaves a floor of $\sqrt{\rho}$ times the fluctuation of a single node. With a correlation of one half, five nodes already give about 0.77 of the single-node relative fluctuation, ten thousand nodes give about 0.71, and the limit is about 0.71, so growing from five nodes to ten thousand improves the measure by only about nine percent. The result holds for nodes of equal variance and a common nonnegative correlation, and real capacities will depart from both assumptions. Only capacity that is weakly or negatively correlated smooths the total, which means nodes in different climates, in opposite hemispheres whose seasons are out of phase, or in different time zones whose daily demand peaks occur at different hours. Nodes in many time zones can offset the

daily cycle, though seasons stay aligned within a hemisphere, and the distribution of people and equipment is strongly weighted to the north. The smoothing available from geography should therefore be estimated from actual correlations and not taken from the number of participants.

14.3 The Grid Behind the Heater

The network's demand for electricity is also geographic, and it has a consequence for the electrical system that is not captured by the energy accounting. Proposition 7.2 shows that a computing heater consumes the same electricity as a resistance heater meeting the same demand, so replacing resistance heating adds no load. Replacing a heat pump is different. A heat pump with coefficient of performance γ draws $1/\gamma$ units of electricity per unit of heat, so a computing heater replacing it draws γ times as much. During a cold spell, when the grid is under the most strain, a large population of computing heaters displacing heat pumps would raise peak electrical demand by that factor. The peak, which sets the capacity that must be built, and not the annual energy, is the relevant external cost, and an assessment of the network must include it. The admissibility rule of Chapter 7 also prevents the network from chasing cheap electricity at times when the household does not need heat, so the flexibility it offers the grid is limited to the timing allowed by thermal storage.

14.4 Jurisdiction and Connectivity

Data and computation are subject to the laws of the places in which they occur. Tasks that involve personal data, or data whose storage or processing is restricted to particular jurisdictions, can be assigned only to nodes in the permitted places. Writing J_j for the set of jurisdictions permitted for task j, the nodes eligible for it are those located in J_j, and the network's effective capacity for that class of task is the capacity within the permitted set, which can be much smaller than its total. The network must carry jurisdiction constraints in its task descriptions of Chapter 6 and treat them as hard restrictions on matching, not as preferences.

Connectivity varies in ways that matter. Household connections are often asymmetric, with much less capacity for upload than for download, and results must be returned over the slower direction. For most of the tasks that suit the network, outputs are small relative to inputs, so the constraint is mild, but for tasks with large outputs it can decide whether a node is usable at all. Rural regions with the greatest heating demand can have the weakest links.

14.5 Placement as an Optimization

Taken together, the placement of divisible work is a transportation problem. Let $x_{ij} \geq 0$ be the amount of task j placed at node i, let d_{ij} be a measure of the distance between node i

and the data of task j, and let K_i be the capacity of node i. The scheduler minimizes total attributable energy plus transport,

$$\min_{x} \sum_{i,j} x_{ij} (e_i C_j + \epsilon D_j d_{ij})$$

subject to each task being fully placed, node capacities, and the feasibility restrictions on latency and jurisdiction, which delete the variables x_{ij} that violate them. This is a minimum-cost flow problem and can be solved efficiently when work is divisible. It gives a reference value against which a practical heuristic can be judged, in the same way as the matching formulation of Chapter 5, and the cost coefficients e_i and ϵ are the quantities that Experiments A and B must supply.

14.6 Who Bears the Placement

Where computation goes is also a question of who bears its burdens and who receives its benefits. Computational capacity located in cold regions, in the homes whose occupants need heat, places wear and risk in those homes, and a network that compensates them poorly turns the geography of heating demand into a geography of extraction. The households whose heat is already electric, which are least able to switch fuels, are the easiest to enlist and may also be among the least able to refuse, which raises the concern about conditionality discussed in Chapter 17. The proposition that computational geography need not be determined exclusively by the requirements of dedicated facilities has a counterpart. It need not be determined by the economic weakness of the participants either, and the compensation and consent rules of the earlier chapters apply to the choice of where to build the network as well as to how it operates.

14.7 Conditions Under Which the Proposal Fails

The geographic argument fails when the work that is valuable to the network is too tied to data or too sensitive to delay to be moved, when the correlation of capacity across nodes is high enough that no practical aggregation smooths it, when the added peak load on the grid outweighs the energy savings, when jurisdictional restrictions fragment the network into pieces too small to be useful, or when the compensation needed to enlist participants in the places where heat is needed exceeds the value of the work. In any of these cases the placement problem has no profitable solution, and the architecture would be confined to local uses.

Chapter 15

The Long Horizon of Efficiency

This chapter develops the idea that the network may require decades to realize its advantages. Early operation may involve substantial discovery costs, coordination failures, unreliable hardware, and inefficient task placement. The claim is easy to make and hard to test, because a sufficiently long horizon can excuse any early failure. The task of the chapter is therefore to say precisely what would have to be true for a slow start to be worth it, to separate the several senses in which a project can be said to break even, to show how sensitive the conclusion is to the assumptions, and to identify the measurements that could refute it.

15.1 Instantaneous and Lifetime Efficiency

Ordinary efficiency evaluates a system at a moment, through a ratio such as

$$\eta(t) = \frac{\text{useful output at } t}{\text{resources consumed at } t}.$$

That ratio does not capture the value of an architecture that becomes more capable as it learns about its own participants. A discounted lifetime assessment can be written

$$V_T = \sum_{t=0}^{T} \frac{B_t - C_t}{(1+r)^t},$$

where B_t represents verified benefits, C_t represents all attributable costs, and r is the discount rate. Two architectures are compared by the difference in their values V_T, not by their efficiencies at any single date. A centralized system typically has a high $\eta(0)$ and improves slowly. A network of reused resources may have a low $\eta(0)$ and a higher asymptote, and the question is whether the difference in asymptotes pays for the difference at the start.

15.2 History as a Productive Resource

As operating history accumulates, the system can learn which resources are available, which combinations work, and which experiments are worth repeating. Formally the network accumulates a history

$$H_t = \{e_1, e_2, \ldots, e_t\},$$

in which each event records something learned, such as a successful allocation, a failed experiment, a hardware limitation, or a newly discovered capability. The allocation policy is revised from that record,

$$\pi_{t+1} = \mathrm{Update}(\pi_t, H_t),$$

so that the accumulated history is itself a productive resource. A device that joins, fails, moves, or returns does not force the system to rediscover everything. What is learned includes the availability pattern of each node by hour and season, its reliability, the tasks it handles well, and the experiments already tried. History also has costs. It must be stored and protected, it can become stale as hardware and households change, and it can be privacy-sensitive, so the policy that uses it must include a way of forgetting, such as weighting recent events more heavily than old ones.

Learning which of many unreliable nodes to use is a problem with a known structure. Assigning work to nodes whose reliability is unknown and must be discovered by use resembles the multi-armed bandit problem, and for stationary reward distributions there are algorithms whose cumulative regret, the shortfall relative to always choosing the best node, grows only logarithmically with the number of rounds [3]. The cost of learning then grows slowly while the benefit of having learned accrues in proportion to the volume of work, which is the formal basis of the intuition that early inefficiency can be repaid. The guarantee requires stationarity, however. Nodes arrive and depart, seasons change, and hardware ages, and under such drift the regret can grow linearly unless the policy keeps exploring at a rate matched to the rate of change. The logarithmic result is a best case and the actual cost of learning in a drifting network has to be measured.

15.3 A Learning-Curve Model

Empirical studies of manufacturing and operations have often found that the cost of a unit falls as a power of cumulative output, a regularity known as Wright's law [41]. Take it as a model of accumulated operational knowledge. Let the cost per task after x cumulative tasks be

$$c(x) = c_0 \, (x_0 + x)^{-\alpha},$$

with exponent $\alpha > 0$ and an initial experience stock $x_0 \geq 0$ that represents knowledge carried over from related systems, so that the cost is finite at the start. Let each completed task have verified value v, and let k tasks be completed per period. Approximating the cost of every task in period t by that of the unit at cumulative volume kt, the net benefit of

period t is

$$n_t = k\big(v - c_0(x_0 + kt)^{-\alpha}\big), \qquad t \geq 1.$$

Write $X^* = (c_0/v)^{1/\alpha}$ for the cumulative experience at which a single task first costs no more than its value.

The phrase break-even covers three distinct events, and the chapter's conclusions depend on keeping them apart. Marginal break-even is the point at which an individual task becomes profitable. Cumulative break-even is the point at which the project has recovered all the losses it ran up before that. Discounted break-even is the point at which the value, weighted by the discount rate, becomes positive. They occur in that order and can be separated by long intervals.

Proposition 15.1. (a) Marginal. *The net benefit n_t is negative before and nonnegative after $t_m = (X^* - x_0)/k$, so individual tasks become profitable at cumulative volume X^*.*

(b) Cumulative. *In the continuous approximation with $x_0 = 0$ and $\alpha < 1$, the cumulative net benefit $N(T) = \int_0^T n(t)\,dt$ is nonnegative exactly when*

$$kT \;\geq\; X^*\,(1-\alpha)^{-1/\alpha}.$$

(c) Discounted. *Fix a finite horizon T and suppose $t_m > 1$, so that $n_1 < 0$, and that $N_T = \sum_{t=1}^{T} n_t > 0$. Then there is a unique discount rate $r^* > 0$ such that $V_T(r) = \sum_{t=1}^{T} n_t (1+r)^{-t}$ is positive if and only if $r < r^*$.*

Proof. (a) The inequality $n_t \geq 0$ is equivalent to $(x_0 + kt)^{\alpha} \geq c_0/v$, which gives $t \geq t_m$.

(b) With $x_0 = 0$, $N(T) = kvT - c_0 \int_0^T k\,(kt)^{-\alpha}\,dt = kvT - c_0\,(kT)^{1-\alpha}/(1-\alpha)$. The condition $N(T) \geq 0$ is equivalent to $(kT)^{\alpha} \geq c_0/\big((1-\alpha)v\big)$, which rearranges to the stated bound.

(c) Let $z = 1/(1+r) \in (0,1)$ and $P(z) = \sum_{t=1}^{T} n_t z^{t-1}$, so that $V_T = zP(z)$. Since n_t is increasing in t and $n_1 < 0 < N_T$, the coefficient sequence has exactly one sign change. By Descartes' rule of signs, P has exactly one positive root. Because $P(0) = n_1 < 0$ and $P(1) = N_T > 0$, that root z^* lies in $(0,1)$, and $P(z) > 0$ exactly for $z > z^*$. Setting $r^* = 1/z^* - 1$ gives the claim. □

Part (b) shows that cumulative break-even requires more volume than marginal break-even by the factor $(1-\alpha)^{-1/\alpha}$. This factor is about three for an exponent of one sixth and about 3.3 for the exponent of 0.32 that corresponds to a twenty percent cost reduction per doubling of volume. A discrete calculation agrees with the continuous approximation closely, for example locating the cumulative break-even within a tenth of a percent for $\alpha = 0.2$ and $c_0/v = 10$. Part (c) replaces what would otherwise be an unsupported claim that every learning-based project has a critical discount rate. The rate exists and is unique under stated conditions, and it can fail to exist, for instance when the cumulative sum over the horizon is never positive.

The thresholds for the learning exponent follow. For marginal break-even by the horizon, the last task, at cumulative volume $x_0 + kT$, must cost no more than its value, which requires

$$\alpha \;\geq\; \frac{\ln(c_0/v)}{\ln(x_0 + kT)}.$$

This is the threshold for the final task and not for the project. For cumulative break-even with $x_0 = 0$ the requirement from part (b) is

$$\alpha \ln(kT) + \ln(1 - \alpha) \;\geq\; \ln(c_0/v),$$

which is implicit in α and has to be solved numerically. If the initial cost is ten times the value of a task and the network expects to complete 10^6 tasks over its planning horizon, the exponent must be at least one sixth for the last task to pay for itself, and about 0.18 for the project to have recovered its losses. These are numbers a pilot can estimate, and a measured exponent below the second refutes the hypothesis for that horizon.

Two qualifications follow, and both cut against the hypothesis. First, the pure power law has no floor, whereas real costs approach a limit. If the cost has the form $c(x) = c_\infty + (c_0 - c_\infty)(x_0 + x)^{-\alpha}$, the network can succeed only if $v > c_\infty$, so learning cannot rescue an architecture whose long-run cost exceeds the value of its output. Second, the break-even time falls as the volume k rises. A large centralized operator completes more tasks per period and therefore moves down its learning curve faster in calendar time than a distributed network of modest contributions. The advantage of the network cannot be speed of learning. It has to be a lower asymptotic cost or a lower cost of capital, which is the argument of Chapter 13, and the model should be applied to both architectures and not only to the one under study.

15.4 Irreversibility and the Option to Stop

One genuine asymmetry favors the network under uncertainty, though it is easy to overstate. A data center commits large sums to assets that cannot be recovered if the venture fails, whereas the network grows by small voluntary steps using hardware that already exists and is mostly valuable for other purposes. When the future is uncertain, the ability to stop cheaply has value, a point made in the theory of investment under uncertainty [13]. A project whose downside is bounded by small losses can be continued until its prospects are clear, and a project whose losses are sunk cannot.

The advantage depends on reversibility being real, and the cost of participating must be distinguished from the cost of organizing participation. The network may require very little hardware commitment while accumulating substantial institutional commitment: software development, coordination protocols, security infrastructure, verification, recruitment of participants, and contractual and regulatory obligations. These costs are largely sunk and do not shrink when a single household withdraws. The accurate statement is therefore narrower

than that the network is reversible. Its hardware commitment is low and its organizational commitment may be high, and the option value is the value of being able to stop before the organizational investment has grown too large. A research program can exploit this by proceeding in small informative steps, each of which buys information about viability at a cost that can be written off.

The same horizon carries risks that no model of the network's own learning captures. Hardware improves, so the comparison baseline moves. Institutions change and the demand for the network's products may disappear. A twenty or thirty year assessment is a scenario analysis and not a forecast, and it should be reported as a range across plausible discount rates and learning exponents, with each of the three break-even times shown for every combination, and not as a single figure.

15.5 Falsifiability and Experiment D

The hypothesis is not that evolutionary infrastructure inevitably becomes efficient. It is that accumulated operational knowledge can create advantages invisible to short-term utilization metrics, and it is falsifiable by the measures introduced above. It fails if the estimated learning exponent falls below the cumulative threshold for the planning horizon, if the asymptotic cost exceeds the value of the output, if no critical discount rate exists or it lies below any rate that is reasonable to apply, or if a centralized baseline, subject to the same model with its own faster learning, dominates at every date of interest. A more modest and defensible statement of the proposition is that a distributed computational ecology may be worth investigating even when its expected operating advantage is uncertain, because it can purchase information about its own viability through small and reversible commitments.

Experiment D in Appendix B is the means of testing it. It simulates years of heterogeneous node arrivals, departures, failures, and seasonal demand, applies history-aware scheduling alongside centralized and naive distributed baselines, and reports the lifetime value V_T, the cumulative regret of the scheduler, the estimated learning exponent, and the three break-even times, each across a grid of discount rates. The simulation can show whether the mechanism works in principle. Only a field pilot can supply the empirical exponent that decides whether it works in practice.

Part V

Governance, Goodhart, and Selective Continuation

Chapter 16

Why the Network Must Not Reward Raw Computation

This chapter returns to Goodhart's Law and turns the diagnosis of Chapter 3 into design rules. If participants are rewarded simply for the amount of compute they contribute, the network reproduces the competitive escalation described in Chapter 4. Proposition 3.2 showed that in a contest among many identical participants, where a fixed reward is divided in proportion to expenditure, nearly the whole value of the reward is dissipated in spending that adds nothing to the stock of useful results. In the notation of Proposition 3.1, a reward on raw compute makes the proxy the cheapest route to income for every participant who can buy hardware, and the network would then cease to be an architecture for productive reuse and become another mechanism for extracting computational expenditure.

16.1 A Taxonomy of Gaming

It helps to list the ways a participant can raise rewards without raising value, since each calls for a different defense. A participant can purchase hardware beyond what its host needs, or run existing machines continuously, which raises measured contribution at the cost of the host's purpose. A participant can manufacture artificial workloads that are cheap to produce and meaningless to complete. A participant can misrepresent its availability or capabilities in the resource descriptor of Chapter 5. A participant can fabricate results without performing the work, multiply trivial tasks, or post tasks to itself and collect the reward for solving them. A participant who reports energy savings can manipulate the baseline against which savings are measured. Finally, a participant can collude with a verifier, which defeats the check that was meant to catch all of the above. Rewards tied to verified outcomes close the first two openings and leave the rest, and the remaining sections address them in turn.

16.2 Verification

The standard defense against fabricated results is redundant execution. The BOINC platform [2] issues each work unit to more than one volunteer and accepts a result only when independent copies agree. The method works because a result produced by honest computation is reproducible, and fabricated results are unlikely to agree unless the fabricators coordinate.

Proposition 16.1. *Assume that each task is assigned to r replicas chosen independently and uniformly at random from a fixed population, that a fraction q of the nodes collude, that each node holds a single identity, and that the colluders can neither observe nor influence the assignment. The probability that every replica of a given task falls on a colluding node, so that a fabricated result can be agreed upon without detection, is then q^r.*

Proof. Each replica is colluding with probability q, and the assignments are independent, so all r are colluding with probability q^r. If at least one replica is honest, a fabricated result from the colluders disagrees with the honest one and is detected. □

The four assumptions carry the result, and each can fail. If a participant can hold many identities, q is no longer a fixed fraction of the population and can be raised by registering more nodes, so replication must be paired with a cost of identity such as the tying of each node to a physical host. If assignment is correlated, for example because nearby nodes tend to receive the same tasks, the replicas are not independent. If colluders can predict or steer the assignment, the bound q^r does not apply at all. The proposition is a statement about a protocol whose sampling is genuinely random and whose identities are scarce, and it should be read as the specification of those requirements and not as a guarantee that holds in their absence.

The cost of this protection is a factor of r in computation, which the network must set against the value of the work. Where the honest result is expensive to produce and cheap to check, spot-checking is a more economical complement. A fraction of tasks are seeded with problems whose answers are known to the verifier in advance, so that a participant cannot tell which results will be inspected.

Proposition 16.2. *Assume that honest work costs a participant c per task and is always accepted, earning reward G. Assume that a fabricated result avoids the cost c, is inspected with probability s independently of the participant's behavior, is detected with certainty when inspected, and that detection forfeits both the reward and a stake F that would otherwise be returned. Then fabrication is unprofitable if and only if*

$$s \geq \frac{c}{G + F}.$$

Proof. Honest work yields $G - c$. Fabrication yields $(1 - s)G - sF$. Fabrication does not pay exactly when $(1 - s)G - sF \leq G - c$, that is, when $c \leq s(G + F)$. □

The threshold is specific to this payoff model. If detection forfeits the reward but the stake is merely returned, so that $F = 0$, the condition becomes $s \geq c/G$. If detection forfeits only the stake and leaves the reward in place, fabrication yields $G - sF$ and the condition becomes $s \geq c/F$. If inspection is imperfect, the effective s is the product of the inspection probability and the probability that an inspected fabrication is recognized. The proposition shows that the inspection rate needed for deterrence falls as the penalty grows, which suggests large stakes. The stake carries its own problem, though. A required deposit favors participants who have capital, and it reintroduces the advantage of wealth that the network was designed to avoid. The choice between a high inspection rate and a high stake is a choice about who is excluded, and it should be made with that in mind. Physical experiments add further tools, namely replication at independent sites, calibration against reference standards, and blind samples whose properties the verifier knows, as described in Chapter 8.

16.3 Bounding Participation

Verification addresses fabrication and leaves the arms race itself, which is addressed by bounding what can be rewarded.

Proposition 16.3. *Suppose the rewarded quantity of participant i is $\min\{x_i, \bar{x}_i\}$, where x_i is the quantity it produces at a positive marginal cost and $\bar{x}_i$ is a cap fixed by its host system, and the reward pool is divided in proportion to the rewarded quantities. Then producing more than $\bar{x}_i$ is strictly dominated, so no participant gains from raising the rewarded quantity above its cap, and the total rewarded quantity is at most $\sum_i \bar{x}_i$.*

Proof. Spending above $\bar{x}_i$ does not change the rewarded quantity $\min\{x_i, \bar{x}_i\}$, and therefore does not change the share received, while it raises cost. Any strategy with $x_i > \bar{x}_i$ is dominated by the same strategy with $x_i = \bar{x}_i$. □

The cap is the mechanism by which the purpose of the host system limits the contest. In a thermal installation, for instance, the cap on rewarded power is the heating demand $D_i(t)$ of Chapter 7, so that additional hardware earns no additional reward for raising the reported quantity. The result is narrow. It removes the marginal reward for additional reported quantity and does not remove every incentive for capital expenditure. A participant may still buy more efficient hardware to lower the cost of reaching the cap, or more reliable hardware to reach it more consistently, and those investments are not harmful in themselves. The result also depends on the cap being fixed by the host and verifiable by others, which a participant who can misreport the host's demand could attack. It is a design choice that Bitcoin-style rewards lack, because there the only limit on expenditure is the value of the reward itself. A cap also reflects the larger principle that capacity should be tied to the primary function of the participating system and not to the participant's ability to pay.

16.4 Gates and Fines

The rule that the network must never reward participants for defeating the constraints that justified their participation needs a particular form. A fine for violating a constraint can be priced in. If violating gives a participant a gain g and a fine φ is imposed with probability s, the participant violates whenever $g > s\varphi$, and with a low detection probability or a high gain the violation pays. A gate does not have this weakness. Under a gate the reward is

$$R_i = G \cdot \mathbf{1}[\text{all constraints satisfied}] \cdot V_i,$$

with V_i the verified outcome, so a detected violation sets the reward to zero however large the gain from violating inside the reward mechanism. A gate limits what is rewarded and does not by itself guarantee compliant behavior. Its effect depends on detection and enforcement, and on whether violation earns the participant benefits outside the reward mechanism. A participant who can conceal a breach, or who profits from it elsewhere, is unaffected by the gate. What the gate achieves is to keep the prohibited action out of the set of rewarded outcomes, so that the constraint is not offered to the participant as an item on a price list. Constraints that protect the host, such as safety limits, thermal envelopes, and privacy boundaries, should be implemented as gates and not as fines.

16.5 Provenance, Independence, and Review

Provenance records, in the append-only form of Chapter 6, make results traceable to the configuration, inputs, hardware, and participant that produced them, so that later audits can reconstruct how a reward was earned. Verification must be independent of the beneficiaries. A verifier who depends on the network for income, or who is selected by the parties being verified, is subject to the same pressure that shapes what an investigator notices, discussed in Chapter 2. Verifiers should therefore be drawn at random, rotated, and paid in ways that do not depend on the results they approve.

Metrics also decay. Campbell's principle [12] predicts that any indicator used to allocate rewards will be corrupted, so the performance criteria must be reassessed on a schedule and replaced when they have been gamed. Test tasks and inspection rates should be changed unpredictably, and each criterion should carry a stated expiry, after which it is retired unless a review renews it. The network should expect to rewrite its own measures regularly and should not treat any as permanent.

16.6 The Governing Rule and Its Limits

The most important institutional rule is that the network must never reward participants for defeating the constraints that justified their participation. The mechanisms of this chapter, verification, caps, gates, independence, and review, are ways of implementing it.

None of them removes the underlying pressure. Even verified-output metrics can be gamed. Scientific results can be fabricated, trivial tasks can be multiplied, and reported energy savings can depend on manipulated baselines. The aim is therefore not a reward scheme that is immune to gaming but one in which the cost of gaming is high, the benefit bounded, and the discovery of gaming expected, so that the network corrects itself after each failure.

Chapter 17

Local Authority and the Right to Remain Idle

This chapter addresses ownership, autonomy, and refusal, and it supplies the definition of permission on which the earlier chapters depend. Chapters 7, 10, 12, and 16 each assume that some set of actions is permitted and that rewards operate inside it. Unless that set is defined independently of the network, the reward mechanisms become the very authority they were meant to constrain. A participating computer remains the property and instrument of its owner. A furnace remains a heating appliance. An air conditioner remains a cooling appliance. The computational network receives limited permission to use residual capabilities, and it does not acquire authority over the primary function of the device.

17.1 A Hierarchy of Authority

Permission is built in two layers whose sources differ. The first is the safety envelope $\mathcal{S}_i(t)$ of device i, the set of actions compatible with its certified operating limits, enforced by independent controllers. It is fixed by certification and by the manufacturer's limits, and no party in the network can enlarge it, the owner included, so far as the network's interface is concerned. The second is the owner policy $\mathcal{O}_i(t)$, the set of actions the owner currently allows. It changes only by the owner's own act and is revocable at any time. The permitted set is their intersection,

$$\mathcal{A}_i(t) = \mathcal{O}_i(t) \cap \mathcal{S}_i(t).$$

Let $\mathcal{R}_i(t)$ denote the actions the network would like to perform. The executable set must satisfy

$$\mathcal{E}_i(t) \subseteq \mathcal{A}_i(t) \cap \mathcal{R}_i(t).$$

The inclusion has a direction that matters. The network's preferences select within the permitted region and never determine it.

The relation to rewards is then stated exactly. A reward schedule offered by the network

is an input to the owner's decision about $\mathcal{O}_i$, since an owner may choose to enlarge the policy in return for payment, and it is never an input to $\mathcal{S}_i$ or a means of bypassing $\mathcal{O}_i$. The network cannot enlarge $\mathcal{A}_i$ by offering more, because there is no channel through which an offer acts on the set other than the owner's own decision, and the safety envelope cannot be bought by anyone. The optimization problem of Chapter 16 is therefore posed on $\mathcal{A}_i(t) \cap \mathcal{R}_i(t)$. Rewards are defined for actions inside the region, and actions outside it carry no reward at all.

17.2 Revocation

The right to refuse would be empty if it could be overridden by commitments already entered into. It is therefore specified as a protocol with four requirements. The executable set is recomputed continuously from the current $\mathcal{O}_i$. Obligations to the network are conditional on $\mathcal{O}_i$ and carry no liability to the owner beyond the loss of future rewards and of eligibility for work that depends on the withdrawn capacity. Wind-down actions, collected in a pre-certified set $\mathcal{W}_i \subseteq \mathcal{S}_i$ that includes checkpointing, controlled cooling, and shutdown, can be carried out by the local controller without the network's cooperation. And each of those actions completes within a certified latency τ_i.

Proposition 17.1. *Under these four requirements, if the owner withdraws permission at time t_0, then during $[t_0, t_0 + \tau_i]$ the only actions executed are those in $\mathcal{W}_i$, and after $t_0 + \tau_i$ no action is executed at the network's request, whatever obligations remain outstanding.*

Proof. After t_0 the owner policy $\mathcal{O}_i$ contains no action requested by the network, so by the first requirement the executable set contains none. Operations already in progress are brought to rest by the wind-down actions, which by the third requirement run locally and by the fourth complete within τ_i. By the second requirement, outstanding obligations create no claim on the device that execution could satisfy. □

The proposition is a specification of a protocol and not a law of nature, and it should be read that way. The proof shows that the four requirements together are consistent and sufficient. It does not show that any real system meets them. What makes the property true in practice is enforcement in the local controller, which must not depend on a network that has an interest in continuing the computation.

The latency τ_i marks the difference between immediate revocation of future work and the physical impossibility of stopping every operation instantly. A heating module cannot cool at once, and a liquid-cooled loop may need to circulate for a time to remove residual heat safely. Those steps belong to the safety envelope and not to the network's claim on the device. The loss from interrupted work falls on the network and not on the owner. Interrupted tasks must be checkpointable, as required in Chapter 7, and results lost in a revocation are a risk the network bears for having relied on voluntary capacity.

17.3 Local Priority

The same ordering governs the use of resources at every moment. The primary function and safety of the device come first, the owner's own use second, and the network's work third. Let $c_i(t)$ be the total capacity of the device, $c_i^{\text{primary}}(t)$ the capacity required by the primary function, and $c_i^{\text{owner}}(t)$ that required by the owner's own use. The capacity available to the network is

$$c_i^{\text{net}}(t) \leq c_i(t) - c_i^{\text{primary}}(t) - c_i^{\text{owner}}(t),$$

evaluated at the time of use, not at the time a contract was made. This is the form taken by the rule of Chapter 5 that contribution percentages apply to what remains after local requirements are met, and it is preemptive. When local demand rises, the network's allocation falls in the same instant.

17.4 Privacy and Isolation

The network should learn only what it needs in order to assign work. The resource descriptor $r_i(t)$ and the results of tasks are the declared observables, and everything else about the device and its household is outside the network's view. Workloads run in isolation from the owner's own data and network.

Thermal scheduling raises a problem that is easy to overlook. The heating demand $D_i(t)$ of Chapter 7 is a record of when a building is occupied, when it is vacant, and how it is used, and it is personal data. A scheduler that receives raw demand profiles can infer household routines. The design should therefore transmit coarse availability windows, aggregated or perturbed, and keep the fine-grained demand local, where the admissibility condition is enforced by the device's own controller without reporting its inputs.

17.5 Availability Is Not Authorization

A device may have unused capacity without that capacity being available to the network, and the point deserves stating plainly. Physical availability describes what the hardware can do. Legitimate authorization describes what its owner has agreed to. The first never implies the second. Idle resources are not unclaimed resources. The network does not own the idle capacity it discovers, and discovery confers no title to what was found.

This returns to the distinction of Chapter 1 between productive and wasteful idleness. A machine kept idle may be serving its owner's option to act at once, or the owner's preference for privacy and control. The owner is not required to show that the idleness is productive, and refusal needs no justification. The right to remain idle is therefore not a technical defect to be engineered away but a governance principle. It answers the efficiency doctrine examined in Chapter 2 directly: the existence of an optimizable resource does not establish an obligation to optimize it.

17.6 How the Right Erodes

A right that holds in the protocol can be lost in the setting around it. Default settings that enroll households unless they opt out turn consent into inertia. Bundling participation with an essential service makes refusal costly. Subsidies that are available only to participants convert an offer into a pressure. Social and reputational expectations can do the same. Each of these leaves the formal permission intact while making its exercise hollow.

Three rules guard against it. Participation must be opt-in, and the primary service of the host, whether heat, cooling, or ordinary computing, must not be conditioned on participation. Revocation must carry no penalty beyond the loss of future rewards. And permission must expire unless renewed, so that consent given once is not treated as consent in perpetuity. The first of these may be called the no-conditionality rule, and it is what distinguishes a network that accepts what owners freely offer from one that extracts what its economics make hard to withhold. A fourth test is the dissolution condition of Definition 2.1: whether the network could be dissolved without loss of any participant's primary function, without uncompensated participation costs, and without surviving obligations. A network that fails it has acquired a hold on its participants regardless of what its protocol says about consent.

17.7 Dependence and Reliance

A principle that forbids capture has to say what it forbids, because the network cannot both guarantee that nobody becomes dependent on it and grow into infrastructure worth having. The line is drawn at the host's primary function. Dependence is prohibited when the function for which the device exists, such as keeping a building warm, keeping food cold, or serving its owner's own computing, would be impaired by disconnecting the network. A household that loses its heat when the network goes away is dependent in the prohibited sense. Reliance is permitted when a participant values a service that exists only because the network does, such as access to results, a share of the network's revenue, or diagnostic reports on its equipment, and could do without it at no cost to the function of the device. Voluntary reliance of this kind is the ordinary consequence of providing something useful.

The distinction does not make the underlying tension disappear. The more useful the network's services become, the more reasonably participants rely on them and the stronger the case for preserving the network for their sake, which is a source of institutional momentum. The rules of this chapter, opt-in participation, penalty-free revocation, expiring consent, the no-conditionality rule, and the dissolution reserve of Chapter 2, are meant to keep that momentum from turning into captivity. They cannot prevent the momentum, and a network that succeeds will have it.

17.8 Conditions Under Which the Principle Fails

The principle fails if the safety envelope cannot be enforced independently of the network, if revocation cannot be made effective within a certified latency, if the information the network receives allows inference of private activity, or if the surrounding economics make nonparticipation prohibitively expensive. In each case the formal structure may remain in place while the right it was meant to protect is gone, and the chapter's rules are therefore best read as conditions to be audited and not as properties assured by design.

Chapter 18

Computation as an Ecology of Admissible Contributions

The concluding chapter is a critical synthesis and not a triumphant summary. The monograph began as a proposal. In the course of examining it, the preceding chapters have accumulated mathematical and economic constraints, and its conclusion should be more discriminating than the proposal from which it started. The chapter organizes what has been argued into three epistemic categories: results that hold within the models developed, propositions that are conditional on quantities that must be measured, and conjectures that remain open. It declines to promote any member of the third category into a conclusion.

18.1 The Three Forms and the Unit of Participation

The chapter first recalls what has been synthesized. Digital residual computation exploits spare computational capacity. Thermally integrated computation performs digital work while satisfying heating demand. Physical analog computation uses controlled material processes to produce measurements, evaluations, or computational transformations. These activities are coordinated through a heterogeneous network whose structure develops over time.

The fundamental unit of participation is not a processor, a watt, or a completed task, but an admissible contribution.

Definition 18.1 (Admissible contribution)**.** An *admissible contribution* is a verified activity that produces value without violating the independent purposes and constraints of the participating system.

One model of technological development seeks to concentrate resources, maximize measurable utilization, and continuously expand dedicated infrastructure. Another seeks to discover useful relationships among existing resources, preserve local authority, and accumulate improvements through long-term experimentation. The second is not guaranteed to

outperform the first. Its value must be established through measurement and not asserted as a philosophical preference.

18.2 Results Established Within the Models

The word established is used in a limited sense. The results in Table 18.1 are mathematical or physical statements about the models developed in the preceding chapters. They are not guarantees about the proposed architecture, since each rests on stated assumptions that real systems may violate.

Result	Where	Depends on
Electricity consumed by a computing heater appears as heat in the enclosure, equal to resistance heating for the same input	Chapter 7	Delivered fraction near one; no energy is created
Under the admissibility rule, household electricity use equals that of a resistance heater whatever the workload	Prop. 7.2	Resistance baseline; fallback element supplies the remainder
A proxy reward can fail completely at its optimum, and a reward proportional to expenditure dissipates nearly all its value in competition	Props. 3.1, 3.2	The stated effort and contest models; many identical participants
Correlated node capacities leave a floor on the relative fluctuation of total capacity	Prop. 14.1	Equal variance; common nonnegative correlation
Signal delay over distance bounds latency from below	Chapter 14	Propagation speed in fiber
Exclusive-or combination of independent biased bits shrinks the bias multiplicatively, and one adaptive contributor defeats it	Props. 12.1, 12.2	Independence; what the contributor observes and controls
Without a common reference configuration, configuration and site effects cannot be separated, and cross-site comparison has a variance floor	Props. 9.1, 10.2	Additive site-effect model
Replication, spot-checking, and capped rewards have stated guarantees, and gating removes prohibited actions from the rewarded set without ensuring compliance	Props. 16.1, 16.2, 16.3	Random assignment and single identities; stated payoffs; detection and enforcement

Table 18.1: Results established within the models of the monograph.

Two of these deserve emphasis because they are counterexamples to the most tempting form of the argument. The first is the contrast between resistance heating and heat pumps. A computing heater adds no load when it displaces resistance heating, and when it displaces a heat pump it draws as much as the coefficient of performance times the electricity for the same heat, with the greatest effect at the peak of grid demand. Useful heat recovery therefore cannot automatically be counted as an energy saving, and the relevant baseline is

what the household would otherwise have used. The second is the correlation floor, which shows that adding participants does not by itself smooth the network's capacity.

18.3 Conditional Propositions

The remaining claims of the monograph are conditional, and each names the quantities on which it depends. Residual computation can outperform dedicated infrastructure when avoided capital expenditure and useful secondary outputs exceed the costs of coordination, energy, maintenance, and reliability, and Proposition 13.1 shows that, with the cost structures typical of a data center, the network is cheaper only below a crossover volume. Thermally integrated computation is justified only if the value of the computation covers the additional electricity relative to the baseline appliance together with the added capital, as stated in Proposition 7.1. Physical analog computation is valuable when controlled experiments yield results that are accurate and informative enough to beat simulation or laboratory testing at the same reliability, which depends on discriminability, site variance, and the question being asked, as developed in Chapters 8 and 9. Long-term learning can improve the network's economics, but only when the learning exponent, the cost floor, the discount rate, and the accumulated organizational costs allow cumulative break-even within a horizon of interest, as in Proposition 15.1. The right to remain idle holds in practice only if the safety envelope and revocation are enforced independently of the network, as specified in Proposition 17.1.

Each of these statements can be turned into a measurement. Experiments A through E in Appendix B are designed to supply the quantities, and until they have done so the statements remain conditional in the strict sense that their antecedents are unknown.

18.4 Open Conjectures

Three conjectures are more ambitious, and they are empirical questions that the monograph does not answer. The first is that a heterogeneous network of reused resources becomes economically competitive over decades. It requires a learning exponent and a cost floor that no pilot has yet measured, and it competes against centralized operators that learn faster in calendar time. The second is that geographically dispersed thermal systems provide scientifically valuable experimental capacity. The staged test of Chapter 9 is the means of settling it, and its outcome could be that the network is a heating system with sensors attached. The third is that decentralized governance can resist incentive corruption of the Goodhart kind. The mechanisms of Chapter 16 reduce specific attacks and do not remove the underlying pressure, and only adversarial experience will show whether the residual is tolerable. Two further questions are narrower: whether there is enough demand for diverse-provenance randomness to sustain a physical entropy service, and whether fleet-level diagnosis of cooling equipment is worth the cost of the instrumentation. A philosophical question stands beside these. It is whether an engineered alternative to centralized infrastructure escapes the assumptions of technological civilization or reproduces them in another organizational form,

as the objection of Chapter 2 presses. The dissolution condition of Definition 2.1 makes one part of that question testable, namely whether participants could leave without loss of function, uncompensated cost, or surviving obligation. It does not answer the rest, and the monograph leaves the rest open.

The monograph refuses to promote these possibilities into conclusions. A result on any of them would have to come from the experiments of the appendix and from field pilots, and the failure conditions stated at the end of each chapter are the observations that would count against the hypothesis.

18.5 Status of the Opening Promises

The Preface and the first chapter make several promises. Table 18.2 classifies each as established within the models, conditionally supported, experimentally testable, or unresolved, so that the conclusion can be checked against the introduction.

Promise	Status	Where
Maximizing utilization is not the same as maximizing social value	Established as a conceptual point; the comparison of arrangements is conditional	Chapters 1, 13
Idle capacity can have value of its own	Conditional on measurements of interference and availability	Chapters 1, 5
Residual capabilities can be organized into a useful network	Conditional; testable by Experiment A	Chapters 5, 6, 13
Computation can ride on heating demand without added electricity	Established against a resistance baseline; conditional against a heat pump	Chapters 7, 14
Physical processes can supply valuable measurements	Conditional; testable by Experiment C and a staged field test	Chapters 8 to 12
Metric-driven escalation can be bounded	Model results established; resistance in practice unresolved	Chapters 3, 16
Efficiency can accrue over decades	Conditional; testable by Experiment D and field pilots	Chapter 15
Local authority and the right to remain idle can be preserved	Specification established; enforcement conditional	Chapter 17
The alternative escapes the assumptions of technique	Unresolved; reversibility partly testable	Chapters 2, 18

Table 18.2: Status of the promises made in the Preface and the first chapter.

18.6 Order of Operations

The difference between this architecture and the optimization philosophy of Chapter 2 comes down to an ordering. The conditions of participation are established first, by owners and by

the purposes their systems already serve, and optimization happens afterward and within them. Efficiency is not permitted to determine the conditions under which participation occurs. The network optimizes within a distributed set of permissions, and does not treat every available resource as something to be exploited to the maximum.

The most valuable computational infrastructure may not be the one that performs the greatest quantity of computation, but the one that discovers the greatest quantity of independently useful computation within resources already committed to other legitimate purposes. This changes the question infrastructure is designed to answer. In place of asking how the utilization of computational resources can be maximized, it asks what computations existing physical and technological systems can perform without compromising the reasons those systems exist. That proposition is a hypothesis about what is worth asking, and the chapters above have tested it against its strongest counterexamples without removing them.

18.7 Closing Statement

The argument of this treatise has not been that distributed computation is inherently superior to centralized computation, that unused hardware is free, or that thermodynamic activity becomes computation merely by being observed. Each of these propositions fails under identifiable conditions. The narrower claim is that existing physical and technological systems may possess capabilities that can be recruited for independently valuable work without displacing their primary functions. Whether such recruitment is worthwhile depends on the costs of coordination, verification, energy, maintenance, and the preservation of local authority.

The central problem is therefore not how to extract the greatest possible quantity of computation from the world's existing infrastructure. It is how to discover which contributions remain admissible when the purposes of that infrastructure are treated as constraints and not as obstacles. A household does not exist to cool a processor. A heating system does not exist to perform an experiment. A personal computer does not become a public resource because its owner leaves it idle.

Nor does a distributed network become beneficial merely because it distributes activity. Correlated demand, geographical concentration, unequal bargaining power, and reward manipulation can reproduce the disadvantages of centralized systems in less visible forms. The relevant measure of success is not the number of participating devices, the quantity of computation performed, or the volume of information collected. It is the independently verified value produced after accounting for the burdens imposed on participants and the alternatives displaced.

The remaining hypothesis is consequently both more modest and more demanding than the one with which the investigation began. An infrastructure composed of bounded, reversible contributions may discover productive arrangements that dedicated infrastructure overlooks. Its development may require decades, and it may fail. Its principal advantage is not immunity to failure, but the possibility of learning from failure without requiring every

experiment to become a permanent institution.

Computation need not always begin with the construction of a computer. It may begin with the recognition that an existing system, already performing a legitimate function, can perform another without surrendering the first.

Appendix A

Mathematical Foundations

Four linked models organize the formal argument, in place of a single universal optimization equation. They are kept distinct because each can fail independently of the others.

Admissible Resource Contribution

Each node i has a set $\mathcal{A}_i(t)$ of permitted actions, formed as the intersection of an owner policy and a fixed safety envelope as developed in Chapter 17, so that it reflects local constraints, safety requirements, ownership, and primary function, and every action taken must satisfy

$$a_i(t) \in \mathcal{A}_i(t).$$

Verified Value

A task produces value only when its result satisfies a specified validation procedure. Numerical throughput alone is insufficient.

Lifetime Resource Accounting

Incremental energy, capital costs, hardware replacement, emissions, and the value of useful outputs are compared over a specified horizon, as in the discounted sum of Chapter 15.

Metric Corruption Under Optimization Pressure

The fourth model examines how rewards based on measurable proxies can cause participants to expand resource consumption or manipulate reported performance. Proposition 3.1 is its simplest instance.

A node may be admissible but economically unproductive. A computation may be scientifically valuable but too expensive. An efficient operation may violate local authority.

A seemingly successful reward system may be optimizing a corrupted metric. A strong treatment resists reducing these distinctions to a single scalar objective.

Appendix B

Empirical Program

The monograph is accompanied by a concrete experimental agenda so that its claims can fail. Five experiments are specified. Each is written as a protocol: the question and hypotheses, the design and population, the measurements, the analysis and the decision rules, the threats to validity and safety, and the observations that would count against the hypothesis. The protocols name the quantities that the preceding chapters left as unknowns, and they commit in advance to what each result will mean. The numbers that appear in them are planning figures chosen to show how a calculation would be done, and they are not predictions of what the experiments will find.

B.1 Principles Common to the Experiments

Every experiment is pre-registered before data collection, with its hypotheses, primary measures, analysis, and decision rules fixed in writing and dated. Null and negative results are reported with the same prominence as positive ones. Energy and cost are measured within a stated accounting boundary, and the same boundary is applied to the baseline against which the network is compared, since the conclusions of Chapters 7 and 13 depend on the choice of baseline. Uncertainties are propagated by the method of Chapter 9, and every reported quantity carries its combined uncertainty. Wherever experiments run in several sites, a common reference configuration is run at every site, because Proposition 9.1 shows that without one the effect of the site cannot be separated from the effect of the treatment.

Experiments that involve people or their property observe the rules of Chapter 17 during the experiment itself. Participation is opt-in and revocable without penalty, the owner’s priority is preemptive, consent expires and must be renewed, and the primary function of every device is protected by its independent safety controller. Data and analysis code are released openly with the report, so that the results can be checked and the experiments repeated by others.

B.2 Experiment A: Residual Computing

Question and Hypotheses

Experiment A asks what a heterogeneous population of ordinary computers can contribute without harming its owners, and at what cost. It informs the descriptors and scheduling of Chapter 5, the routing role of Chapter 6, and the comparison condition of Chapter 13. There are three hypotheses. First, that contribution of five to ten percent of spare capacity can be made without exceeding a stated interference bound on the owner's experience. Second, that the marginal energy per verified unit of work on admitted machines satisfies the operating-energy condition $\rho\, e_{\text{old}}^{\text{marg}} < \text{PUE} \cdot e_{\text{new}}$ for some usable subset of machines, with the replication factor ρ taken from the experiment itself. Third, that coordination costs have the linear structure $F + fW$ assumed in Proposition 13.1.

Design and Population

The population is a set of volunteer machines, at least one hundred, recruited to span hardware age, operating system, memory size, and pattern of use. The design is a within-machine crossover. Each machine passes through a baseline period with the network disabled, followed by treatment periods in randomized order with contribution caps of zero, five, and ten percent, separated by washout intervals so that carry-over effects can be assessed. Three workload classes are used: compute-heavy tasks of the Monte Carlo type, storage replicas, and routing-only participation. Results are verified by replication with $r = 2$ and by seeded spot checks as in Chapter 16, and the realized ratio of raw to verified work is recorded as ρ. A modern server runs the same workloads as a comparator.

Measurements

Interference is measured by a standardized interactive probe that times a fixed small task on each machine, with the median and ninety-fifth percentile compared against that machine's own baseline. Power is measured at the wall with plug-in meters in the idle state with the network disabled and in the busy state under load, giving $e_{\text{old}}^{\text{marg}} = (P^{\text{busy}} - P^{\text{idle}})/\theta$ for the measured throughput θ. The comparator's energy e_{new} is measured at the server, and since the facility overhead cannot be measured on a test server, the power usage effectiveness is varied across a stated range as a sensitivity analysis. Availability sessions are logged, so that the exponential assumption behind Proposition 5.1 can be tested against alternatives. The data per task and the compute time are logged to compute the efficiency of Chapter 5. Messages, bandwidth, and operator time are logged to estimate the fixed and per-unit coordination costs. The hit rate of the failure registry described in Chapter 6 is recorded.

Analysis and Decision Rules

The primary analysis is a mixed-effects model with a random effect for each machine. For the interference hypothesis the planning calculation for a paired comparison uses

$$n = \frac{(z_{1-\alpha/2} + z_{1-\beta})^2 \, \sigma_d^2}{\delta^2},$$

where δ is the interference that would be judged unacceptable and σ_d is the standard deviation of the within-machine difference. For a two-sided level of 0.05 and power of 0.8 the leading constant is about 7.8, so if σ_d is twice δ the experiment needs about thirty-two machines per comparison, and more when the analysis is stratified or an equivalence margin is used. The admission rule that the experiment is meant to produce is a threshold on $e^{\mathrm{marg}}_{\mathrm{old}}$, with machines above it excluded, and the threshold is set from the measured relationship between machine age and marginal energy. The coordination model is judged by comparing the fitted linear cost against a quadratic alternative.

Threats to Validity and Safety

Volunteers are not a representative sample, and the interference probe may miss effects that the owners would notice. Power meters have their own error and are included in the uncertainty budget. Machines are never run beyond the contribution cap or against the owner’s priority, and a machine that exceeds the interference bound is withdrawn automatically.

What Would Count Against the Hypotheses

The first hypothesis fails if the interference bound is exceeded for a substantial share of machines at the five percent cap. The second fails if no usable subset of machines meets the operating-energy condition at any plausible power usage effectiveness. The third fails if coordination cost grows faster than linearly in the volume of work. Any of these outcomes limits the network to settings where the unmeasured values of autonomy and local control justify it.

B.3 Experiment B: Computational Heating

Question and Hypotheses

Experiment B asks whether processors integrated into heating appliances deliver heat equivalent to resistance heating while producing verified computation, and under what conditions the arrangement pays. It tests the accounting of Chapter 7 and the peak-load concern of Chapter 14. The hypotheses are that the delivered fraction f is close to one, that under the admissibility rule household electricity use equals that of a resistance heater

(Proposition 7.2), that hardware degradation under the heating duty cycle is acceptable, and that the economic condition of Proposition 7.1 is met for some baseline.

Design and Population

The experiment has two stages. The first is a laboratory stage in an instrumented chamber, with the computing heater, a resistance heater, and where available a heat pump, each run against a controlled thermal load. The second is a field stage in a small number of homes, each fitted with a computing heater module and a resistance fallback element, in which the heating modes alternate over the heating season. Weather is taken into account through heating degree days. A control group of the same processors runs at steady load on a bench, to separate the wear caused by the heating duty cycle from the wear caused by operation as such.

Measurements

In the laboratory the delivered heat is measured by calorimetry on the air stream, $Q = \dot{m}\, c_p\, \Delta T$, and compared with the electrical input to give f, with an uncertainty budget by the method of Chapter 9. In the field the quantities recorded are electricity use in fifteen-minute intervals, the power drawn by the computing hardware $P_i(t)$ and the demand $D_i(t)$ it is capped by, indoor temperature as a measure of comfort, verified tasks completed, and the fraction of heating hours in which the work queue was nonempty, which is the availability of Chapter 7. Hardware wear is tracked through the number of thermal cycles, fan hours, error counts, and failures. Where a home has a heat pump, its seasonal coefficient of performance γ is measured, since Proposition 7.1 depends on it. Coincident peak demand is recorded on the coldest days.

Analysis and Decision Rules

Weather-normalized electricity use is modeled for each mode as a linear function of heating degree days with a random effect for each home. The pure-addition claim is tested as equivalence of the slopes for the computing heater and the resistance heater within a stated margin, since the claim is that the two are the same and failing to find a difference is not evidence of equivalence. The admissibility rule is checked on the logged data, and any interval with $P_i(t) > D_i(t)$ counts as a violation. The value of the computation v is estimated from verified tasks per kilowatt-hour and the value assigned to a task, and the inequality of Proposition 7.1 is evaluated with the measured p, γ, and k, together with its uncertainty. The peak-load effect is reported as the ratio of coincident peak demand in computing-heater homes to the baseline mode.

Threats to Validity and Safety

Homes differ in insulation, occupancy, and thermostat behavior, and alternation over time is confounded with weather, which the degree-day model corrects only in part. The number of homes will be small. All hardware is certified, has independent thermal cutoffs that the scheduler cannot override, and is installed by qualified personnel. The experiment stops in a home at any sign of comfort loss.

What Would Count Against the Hypotheses

The first fails if f is materially below one in practice. The second fails if the slopes differ by more than the equivalence margin. The third fails if the failure rate under the heating duty cycle is substantially higher than at steady load. The fourth fails if the inequality of Proposition 7.1 cannot be met at realistic values of v and k against the baselines households actually use, in which case the arrangement would be limited to homes with resistance heating. A materially higher coincident peak against heat-pump baselines counts as an external cost that the economic assessment must include.

B.4 Experiment C: Physical Analog Calculation

Question and Hypotheses

Experiment C asks whether a controlled physical apparatus can answer a defined question with enough accuracy and at low enough cost to compete with numerical methods, and how much site-to-site variation the method can tolerate. It tests the framework of Chapter 8 and the design principles of Chapters 9 and 10. The hypotheses are that the discriminability d' is large enough for a stated reliability on a benchmark question, that the observed rate of correct selection agrees with Proposition 10.1, that the thermodynamic computational yield η_X exceeds the yield η_{num} of a simulation answering the same question at the same reliability, and that the method tolerates a measurable amount of site variance.

Design and Population

A laboratory flow loop is built with configurable passage geometry, calibrated sensors, and a safe working fluid, operated well within the pressure and temperature limits of residential equipment. A benchmark question is fixed and registered in advance, for instance which of two geometries gives the lower pressure drop at a given flow rate, with candidate answers and a decision rule specified beforehand. The true answer is established independently in a reference measurement of much higher accuracy. A library of fabricated inserts supplies a range of configurations, and a common reference configuration is run in every session. The scale of the apparatus and its operating point are chosen to match the Reynolds number of the target equipment as discussed in Chapter 9, and the report states which similarity groups are matched. To emulate the variation between homes, controlled disturbances are

applied in separate sessions, such as offsets in inlet temperature, added leakage, and changes in filter loading, with magnitudes chosen from field data on residential systems.

Measurements

The measured quantities are differential pressure, temperature, and flow rate, with the combined uncertainty of each derived quantity. Repeated runs give the within-session noise σ, and the disturbance sessions give the apparent site variance σ_b^2. Energy is measured for the apparatus as a whole, including fans, sensors, and data acquisition, and for the numerical comparator, whose wall-power energy is measured over the time it takes to reach the same reliability on the same question.

Analysis and Decision Rules

The discriminability $d' = \Delta\mu/\sigma$ is computed for each pair of configurations. The number of replications needed for a target reliability is planned from Proposition 10.1: for a reliability of ninety-five percent, $z_\delta \approx 1.645$, and if the true difference is half the noise, $\Delta/\sigma = 0.5$, the planning formula requires $m \geq 2z_\delta^2\sigma^2/\Delta^2 \approx 22$ replications. The predicted and observed rates of correct selection are compared across many pairs. The effect of site variance is examined through Proposition 10.2, by comparing cross-session and within-session comparisons and by locating the level of disturbance at which the observed reliability falls below the target. The yields η_X and η_{num} are compared with their uncertainties at a common accounting boundary and a common reliability.

Threats to Validity and Safety

A benchmark that is easy for simulation will understate the case for the apparatus, and one that suits the apparatus will overstate it, so the benchmark is chosen and registered before either is run. Laboratory conditions are more favorable than a home, which is why the disturbance sessions are included. The apparatus operates within rated limits throughout.

What Would Count Against the Hypotheses

The first two hypotheses fail if d' is too small for the target reliability at an affordable number of replications, or if the observed selection rate departs from the prediction in a way that indicates unmodeled error. The third fails if $\eta_X \leq \eta_{\text{num}}$. The fourth fails if the tolerable site variance is smaller than the variance found in the field, in which case the method is suited to validation in controlled settings and not to distributed discovery.

B.5 Experiment D: Long-Horizon Evolutionary Scheduling

Question and Hypotheses

Experiment D asks whether history-aware scheduling can make a heterogeneous network more economical over a long horizon than naive distributed scheduling and than a centralized alternative, under what parameter values, and with what sensitivity to the discount rate. It tests the claims of Chapter 15. Its hypotheses are that a history-aware scheduler accumulates less regret than a naive one in a drifting network, that the learning exponent emerging from the simulated operation meets the requirement for cumulative break-even within the planning horizon, and that the lifetime value of the network exceeds that of a centralized baseline which learns on its own curve.

Design

The experiment is a simulation of many years of operation. Nodes arrive, depart, and fail according to stated processes, and their availability follows daily and seasonal patterns with correlated regional outages as in Chapter 14. Workloads vary over time. Three schedulers are compared: a centralized scheduler operating a dedicated facility with its own cost structure and learning curve, a naive distributed scheduler that assigns work without using history, and a history-aware scheduler of the bandit type with forgetting, as discussed in Chapter 15. The simulator's parameters are taken from the measurements of Experiments A and B where they exist and are otherwise varied over stated ranges, and nothing is tuned to produce a favorable result.

Measurements

The outputs are the cumulative regret of each scheduler, the net benefit series n_t for each architecture, the lifetime value V_T across a grid of discount rates, the three break-even times of Proposition 15.1, and the critical discount rate when it exists. The learning exponent is estimated from the simulated cost series by regression of log cost on log cumulative volume, and compared with the exponent assumed in the simulator's own learning mechanisms, so that learning that emerges from the dynamics can be distinguished from learning that was imposed.

Analysis and Decision Rules

Uncertain parameters are sampled over their ranges, for example by a Latin hypercube design, and results are reported as distributions across the sample and not as point estimates. Each configuration is run with many random seeds, and confidence intervals are reported for every output. The pre-registered decision rules state the fraction of parameter draws in which the history-aware scheduler must beat the naive one by a stated margin in V_T, and the fraction in which the network must beat the centralized baseline, for the hypothesis

to be counted as supported. The simulator is validated by predicting held-out data from Experiment A before it is used for forecasting.

Threats to Validity

A simulation can show that a mechanism works in principle and cannot establish that the real network behaves as simulated. In particular, the learning exponent is an empirical property of a real system, and a simulated exponent reflects the mechanisms that were built into the simulator. The result is therefore a statement about conditions under which the mechanism would pay, to be confirmed or refuted by a field pilot.

What Would Count Against the Hypotheses

The first hypothesis fails if the history-aware scheduler does not beat the naive one under drift. The second fails if the emergent exponent falls below the requirement of Chapter 15. The third fails if the centralized baseline dominates over the plausible range of parameters, which would indicate that the network's value, if any, lies in reasons other than economy.

B.6 Experiment E: Goodhart Stress Testing

Question and Hypotheses

Experiment E asks which reward rules resist manipulation by participants who seek the reward and not the purpose, and how large the residual vulnerability is. It tests Chapter 3 and the mechanisms of Chapter 16. The hypotheses are that rewards for raw compute induce hardware purchases and artificial workloads as Proposition 3.2 predicts, that rewards for verified results shift manipulation to fabrication and duplication, that replication, spot-checking, capped rewards, and gating reduce the profit from manipulation to the levels the propositions of Chapter 16 predict under their assumptions, and that a residual vulnerability remains whose size can be measured.

Design

The experiment compares several reward schemes in a simulated market and then in a controlled game with human participants. The schemes are rewards for raw compute, rewards for verified results, rewards for reported energy savings, and combinations of these, each with and without the protections of Chapter 16. The protections are varied one at a time and together: replication with different replica counts r, spot-checking at different inspection rates s, stakes of different sizes F, caps on the rewarded quantity, and gates on constraint violations. Participants include honest agents, and adversaries with different capabilities: some that can buy hardware, some that can generate artificial workloads, some that misreport availability or baselines, some that fabricate results, some that collude or hold many identities, and some that learn and adapt their strategy to the rules. A red team

of human participants plays against the rules with small stakes, in a game conducted under consent and with no real harm to anyone.

Measurements

The principal measures are the capital that adversaries spend on hardware, the fraction of rewarded work that is artificial, the verified value delivered per unit of reward paid, the false acceptance rate of fabricated results, the profit of adversaries relative to honest participants, and the welfare of the honest participants. The inspection rate at which fabrication ceases to pay is measured directly and compared with the threshold of Proposition 16.2. The probability of undetected fabrication is compared with the value predicted by Proposition 16.1 as the share of colluding and multiple-identity participants is raised, to find the point at which its independence assumptions fail. The rate at which adversaries find new exploits after the rules change is recorded, to evaluate the policy of rotating criteria.

Analysis and Decision Rules

Each scheme is evaluated by the verified value it produces per unit of reward and by the profit of the best-performing adversary strategy found. Adaptive adversaries are given a fixed budget of search, and the profit they reach is reported as a lower bound on what a determined adversary could achieve. The decision rules fix the verified value per reward and the adversary profit that a scheme must meet to be counted as resistant, and the scheme with the best result under the strongest adversary is preferred, not the one with the best average result.

Threats to Validity

Simulated adversaries are limited by the imagination of their designers, and human red teams play with small stakes, so both understate the ingenuity of motivated attackers in a real market. The experiment can show that particular attacks succeed. It cannot show that no attack will succeed.

What Would Count Against the Hypotheses

The third hypothesis fails if adversary profit under the protected schemes exceeds the stated threshold, or if the observed deterrence threshold departs materially from the theoretical one. A finding that adaptive adversaries find profitable strategies against every scheme would show that decentralized governance cannot, by these mechanisms, resist Goodhart-style corruption, and that conjecture of Chapter 18 would be answered in the negative.

B.7 Summary of the Program

Table B.1 collects the experiments, the chapters they bear on, the quantities they supply, and the decisions they inform.

Experiment	Chapters	Quantities supplied	Decision informed
A: Residual computing	5, 6, 13	Interference bound, marginal energy, replication factor, availability model, coordination cost	Which machines to admit; whether reuse beats dedicated infrastructure
B: Computational heating	7, 14	Delivered fraction, equivalence of slopes, wear, value per kilowatt-hour, peak load	Whether and where computing heaters pay
C: Physical analog calculation	8, 9, 10	Discriminability, selection reliability, yield, tolerable site variance	Whether physical modules can compete with simulation
D: Long-horizon scheduling	15, 14	Regret, learning exponent, break-even times, critical discount rate	Whether the long-horizon hypothesis is worth a field pilot
E: Goodhart stress testing	3, 16	Adversary profit, deterrence thresholds, residual vulnerability	Which reward rules to adopt, and whether governance can hold

Table B.1: The experimental program and its connections to the argument.

The experiments are ordered by dependence. Experiments A and B supply the measurements that Experiment D needs for calibration, and Experiment C supplies the accuracy that the field test of Chapter 9 presupposes. Experiment E can proceed at any time and should precede any deployment that pays participants. A result that refutes one of the hypotheses is a result of the program and not a failure of it.

Bibliography

[1] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@home: An Experiment in Public-Resource Computing," *Communications of the ACM* 45(11), 56–61, 2002.

[2] D. P. Anderson, "BOINC: A System for Public-Resource Computing and Storage," in *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, 4–10, 2004.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-Time Analysis of the Multiarmed Bandit Problem," *Machine Learning* 47(2–3), 235–256, 2002.

[4] E. Barker and J. Kelsey, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators," NIST Special Publication 800-90A, Revision 1, 2015.

[5] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *IEEE Computer* 40(12), 33–37, 2007.

[6] I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort, "Pure and Pseudo-Pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp," *Industrial & Engineering Chemistry Research* 53(6), 2498–2508, 2014.

[7] J. Benet, "IPFS: Content Addressed, Versioned, P2P File System," arXiv:1407.3561, 2014.

[8] S. Bonifield, "Sam Altman Says It's Better for RAM to Be in a Data Center Than on Your Desk," *The Verge*, October 30, 2026.

[9] E. Buckingham, "On Physically Similar Systems: Illustrations of the Use of Dimensional Equations," *Physical Review* 4(4), 345–376, 1914.

[10] V. Bush, "The Differential Analyzer: A New Machine for Solving Differential Equations," *Journal of the Franklin Institute* 212(4), 447–488, 1931.

[11] J. Caffier, “What Nathan Fielder’s ‘You Can See Everything’ Taught Me About Tech Journalism,” *Gizmodo*, October 31, 2026.

[12] D. T. Campbell, “Assessing the Impact of Planned Social Change,” *Evaluation and Program Planning* 2(1), 67–90, 1979.

[13] A. K. Dixit and R. S. Pindyck, *Investment under Uncertainty*, Princeton University Press, Princeton, 1994.

[14] J. R. Douceur, “The Sybil Attack,” in *Peer-to-Peer Systems*, Lecture Notes in Computer Science 2429, 251–260, Springer, 2002.

[15] J. Ellul, *The Technological Society*, translated by J. Wilkinson, Alfred A. Knopf, New York, 1964. Originally published as *La Technique ou l’enjeu du siècle*, Armand Colin, Paris, 1954.

[16] C. Fernando and S. Sojakka, “Pattern Recognition in a Bucket,” in *Advances in Artificial Life*, Lecture Notes in Computer Science 2801, 588–597, Springer, 2003.

[17] N. Gómez Dávila, *Escolios a un texto implícito*, two volumes, Instituto Colombiano de Cultura, Bogotá, 1977. A bilingual selected edition is available as *Scholia to an Implicit Text*, translated by R. Pinzón, Villegas Editores, Bogotá, 2013.

[18] C. A. E. Goodhart, “Problems of Monetary Management: The U.K. Experience,” in *Papers in Monetary Economics*, Volume I, Reserve Bank of Australia, Sydney, 1975.

[19] N. Hansen and A. Ostermeier, “Completely Derandomized Self-Adaptation in Evolution Strategies,” *Evolutionary Computation* 9(2), 159–195, 2001.

[20] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, “A Pseudorandom Generator from Any One-Way Function,” *SIAM Journal on Computing* 28(4), 1364–1396, 1999.

[21] A. Jameson, “Aerodynamic Design via Control Theory,” *Journal of Scientific Computing* 3(3), 233–260, 1988.

[22] W. S. Jevons, *The Coal Question: An Inquiry Concerning the Progress of the Nation, and the Probable Exhaustion of Our Coal-Mines*, Macmillan, London, 1865.

[23] Joint Committee for Guides in Metrology, “Evaluation of Measurement Data: Guide to the Expression of Uncertainty in Measurement,” JCGM 100:2008, 2008.

[24] S. Katipamula and M. R. Brambley, “Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems: A Review, Part I,” *HVAC&R Research* 11(1), 3–25, 2005.

[25] H. W. Kuhn, “The Hungarian Method for the Assignment Problem,” *Naval Research Logistics Quarterly* 2(1–2), 83–97, 1955.

[26] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM Journal of Research and Development* 5(3), 183–191, 1961.

[27] L. Ljung, *System Identification: Theory for the User*, second edition, Prentice Hall, Upper Saddle River, 1999.

[28] W. Maass, T. Natschläger, and H. Markram, "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations," *Neural Computation* 14(11), 2531–2560, 2002.

[29] D. Manheim and S. Garrabrant, "Categorizing Variants of Goodhart's Law," arXiv:1803.04585, 2018.

[30] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Peer-to-Peer Systems*, Lecture Notes in Computer Science 2429, 53–65, Springer, 2002.

[31] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[32] S. B. Pope, *Turbulent Flows*, Cambridge University Press, Cambridge, 2000.

[33] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, 2006.

[34] M. Santha and U. V. Vazirani, "Generating Quasi-Random Sequences from Semi-Random Sources," *Journal of Computer and System Sciences* 33(1), 75–87, 1986.

[35] J. Skalse, N. Howe, D. Krasheninnikov, and D. Krueger, "Defining and Characterizing Reward Hacking," in *Advances in Neural Information Processing Systems* 35, 2022.

[36] S. Sorrell, J. Dimitropoulos, and M. Sommerville, "Empirical Estimates of the Direct Rebound Effect: A Review," *Energy Policy* 37(4), 1356–1371, 2009.

[37] M. Strathern, "Improving Ratings: Audit in the British University System," *European Review* 5(3), 305–321, 1997.

[38] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe Exploration for Optimization with Gaussian Processes," in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR 37, 997–1005, 2015.

[39] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent Advances in Physical Reservoir Computing: A Review," *Neural Networks* 115, 100–123, 2019.

[40] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, "Recommendation for the Entropy Sources Used for Random Bit Generation," NIST Special Publication 800-90B, 2018.

[41] T. P. Wright, "Factors Affecting the Cost of Airplanes," *Journal of the Aeronautical Sciences* 3(4), 122–128, 1936.