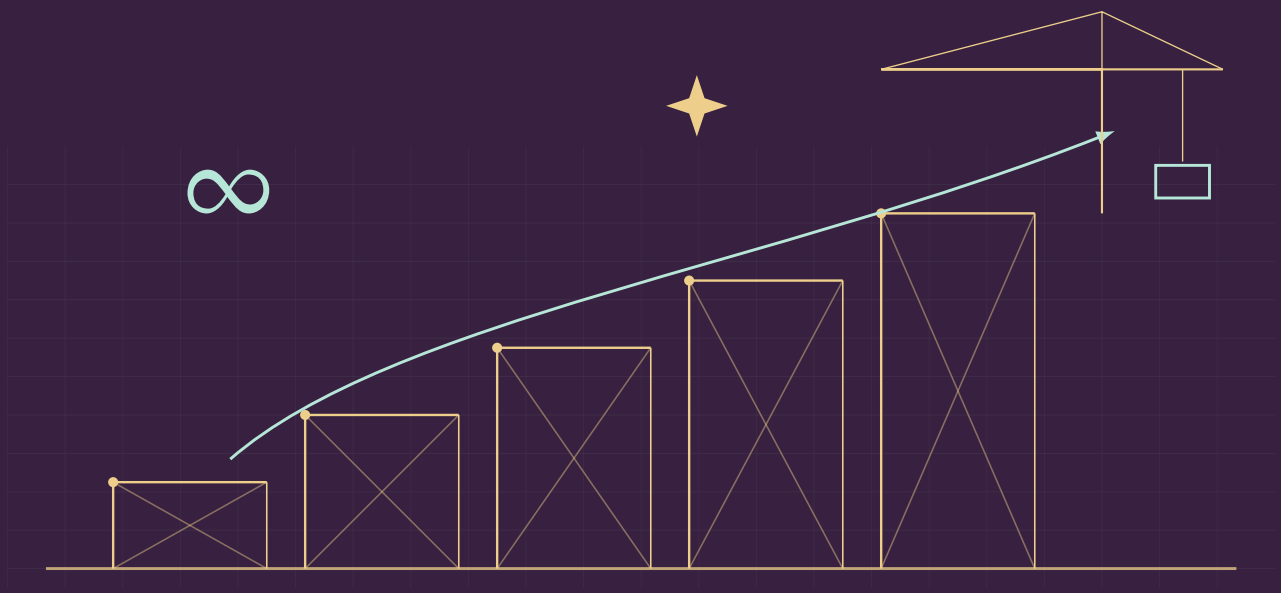
TRANSFORMATIONAL INTELLIGENCE ∞

LEARNING TO BUILD

How Intelligence Learns to Create
What Makes New Possibilities Real



When we build something new,
we change what can be built next.

Vincent Boucher
Montréal, Québec, Canada

ALPHA GENESIS FLYWHEEL Ω
7 November 2026

ALPHA GENESIS FLYWHEEL Ω

LEARNING TO BUILD

How Intelligence Learns to Create What Makes New Possibilities Real

Vincent Boucher · MONTREAL.AI / QUEBEC.AI
Montréal, Québec, Canada · 7 November 2026

ABSTRACT

Intelligence can propose a valuable possibility without possessing the means to realize it. The missing construction may include knowledge, an instrument, a process, resources, and an arrangement that makes their coordinated use worthwhile. We develop *learning to build*: improving the ability to discover and construct these complete, purpose-relative means, then using the evidence to improve fresh construction. The framework extends *Transformational Intelligence*, *Invent the Economy*, and *ALPHA GENESIS FLYWHEEL* Ω to an explicit account of how productive means become executable, maintainable, and reusable. We distinguish the installed frontier from the construction frontier; formalize conditional completeness, complementary prerequisites, and bootstrap obstructions; and separate artifact inheritance from constructor improvement. An executable finite study emits four measurement programs that pass 4,000 input checks. Across 288 supplied resource scenarios, retention can expand or contract the set of attainable deliveries; an equal-content conventional planner ties in every case. These are authored-model results, not measured learning or commerce. A prospective evaluation design isolates retained content, construction procedures, model upgrades, and capital, including tests that remove the donor's finished artifact. The long horizon is a cumulative ability to build scientific instruments, production systems, useful energy capacity, and the means of further construction. Such expansion must earn its physical, economic, and institutional bridges. The central thesis is simple: when intelligence learns to build what a new possibility requires, what it builds can change what becomes possible next.

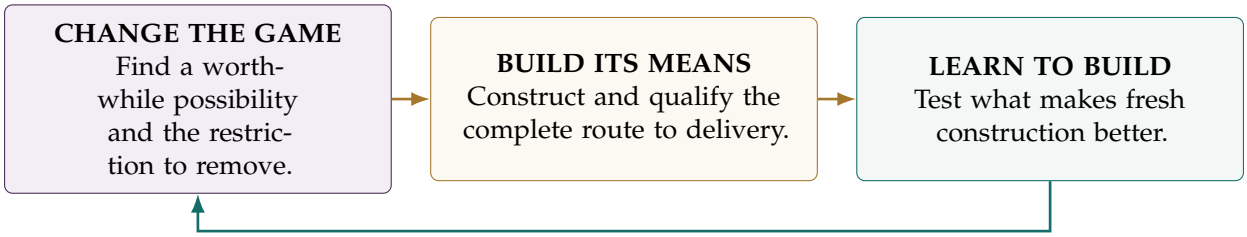
> **Research status.** An operational synthesis, elementary conditional results, an executable finite construction study, and a falsifiable research programme. The study uses authored tasks and assumptions; its generated artifacts and exact checks are real computational outputs. General construction learning, physical deployment, customer acceptance, earned commercial continuation, and civilization-scale expansion require further evidence.

Keywords: transformational intelligence; complete means; capability construction; construction learning; complementary capabilities; successor evaluation; productive capacity.

THE PAPER AT A GLANCE

Build what makes the next breakthrough buildable.

A discovery can tell us what might work. A complete construction gives someone the ability to make it work: the necessary knowledge, instruments, processes, resources, and arrangements, operating together under conditions that can be checked.



THE IMPORTANT DISTINCTION

A useful thing: an instrument performs its intended job.

A useful inheritance: that instrument or its knowledge helps a new job after adaptation.

A better builder: a retained procedure improves the construction of fresh means under a fair comparison.

A better way to improve builders: a learning procedure produces that improvement across fresh campaigns.

These achievements can reinforce one another. None follows automatically from the previous one. A larger collection of tools can hide an unchanged constructor; a better constructor can work with a smaller, more useful collection.

> **The practical question**
> What worthwhile outcome is blocked because the necessary means do not yet work together—and what is the smallest decisive construction that could change that?

What this paper adds. A precise completion boundary; a distinction between using available means and constructing new means; a test for circular bootstrap claims; an executable finite construction witness; and an evaluation design that separates saved artifacts, better procedures, and commercially funded continuation.

Read by purpose. Idea and definitions: Sections 1–3. Construction and evidence: Sections 4–5. Economics and learning tests: Sections 6–7. Industrial horizon: Section 8. Reproduction: Appendix C.

EN FRANÇAIS

APPRENDRE À CONSTRUIRE

Comment l'intelligence apprend à créer ce qui rend de nouvelles possibilités réelles.

Changer les règles du jeu. Construire les moyens qu'il exige. Apprendre à mieux construire. Quand nous construisons quelque chose de nouveau, nous changeons ce qui pourra être construit ensuite. Résumé français; article intégral en anglais.

Contents

1 The next breakthrough may be the ability to build it 4

2 Complete means: an executable whole 5

3 From available tools to a construction frontier 8

4 From a successful build to a better builder 10

5 An executable witness: constructing what the successor needs 12

6 Buildability includes the means of paying for the build 15

7 How to establish that intelligence learned to build 17

8 The horizon: build the means of further construction 19

9 Limits, falsifiers, and the next result worth earning 21

10 Conclusion: the possibility after the possibility 22

A A one-page construction brief 23

B Formal details and interpretation 23

C Reproduction and source lineage 24

D Claim and evidence map 25

1 The next breakthrough may be the ability to build it

Imagine a laboratory that could test an important new material, except that the necessary measurement cannot yet be made reliably. An algorithm can propose the experiment. A designer can draw the apparatus. A manufacturer can make its parts. Yet the useful capability may remain absent: the parts do not interface, calibration is missing, the reference sample is unavailable, or no one can fund the sequence before an accepted result exists. This is an illustrative situation, not an experiment reported in this paper.

The consequential construction is the complete means of performing the measurement. Its output is an ability: someone can now make a distinction, produce a result, or undertake useful work that was previously inaccessible under the stated conditions. The construction can include familiar components. Its significance lies in what they make possible together.

Now consider a second laboratory problem. Keeping the first apparatus may help. Keeping its design principles may help a different apparatus. Learning a better way to identify missing measurements, construct fixtures, diagnose failures, and qualify new designs would be a further achievement. The object that improves has changed from an instrument to the process of making instruments.

Learning to build is improving the ability to turn a worthwhile possibility into working, accepted means—and to make fresh construction better through what was learned.

This is the connective tissue between an ambitious idea and an expanding productive civilization. A useful instrument can improve science. Better science can improve processes. A qualified process can make further instruments, energy systems, or production facilities affordable. Their operation can supply resources and knowledge for another construction. Every connection is a claim to be tested; the sequence is a research and enterprise objective, not an inevitable trajectory.

1.1 The unit of progress is a changed ability

An answer is often exactly what is needed. So is an existing tool, a purchased service, or a simple process change. Construction becomes relevant when an important outcome remains blocked because the required means are absent, incompatible, unreliable, or unaffordable. The right response need not be a larger system. It may remove a requirement that belonged only to the old method, or replace an intricate solution with a mature alternative.

The phrase *what makes new possibilities real* therefore has an operational meaning. State an accepted outcome; identify the initial conditions; construct a permitted route; execute and check it at the claimed evidential level. A digital artifact can be a real computational construction. A simulated factory is a model of a factory. A proposed sale is a hypothesis about a relationship. These distinctions preserve the meaning of each achievement.

We use *worthwhile* to refer to a beneficiary's authorized purpose, with relevant costs, risks, and obligations included. It is not a license for the builder to invent beneficiaries' preferences. The choice of what to make is part of the problem, but acceptance belongs to the people and institutions whose commitments it requires.

1.2 Three foundations, one next question

Transformational Intelligence distinguishes an intended purpose, a formulation for pursuing it, and the instrument used to judge success [1]. A change of representation, observation, dependency, or

coordination can remove a restriction that more search inside the old formulation cannot overcome. The change still needs an executable realization and a bridge back to the purpose.

Invent the Economy makes the productive arrangement itself an object of invention [2]. Technology, participation, prices, risk allocation, and the timing of funding may have to change together. Its separate claims about technical possibility, modeled viability, accepted arrangements, exchange, and repeated adoption remain essential.

ALPHA GENESIS FLYWHEEL Ω connects frontier developments to new valuable possibilities and to improvement of the machinery that discovers them [3]. It already describes complementary capabilities, a construction grammar that must remain open to revision, and causal tests of successor benefit.

The present paper puts a further question at the center:

How does an intelligence become better at constructing the complete means of a new possibility?

Its contribution is an operational synthesis and a sharper unit of evaluation. We define a complete construction witness, distinguish installed and constructible capability, make bootstrap and ongoing obligations explicit, and propose transfer tests in which the donor's finished artifact is unavailable. These ideas develop the trilogy; they do not claim to originate systems engineering, program synthesis, economic complementarity, or learning from experience.

1.3 A bridge between established fields

The engineering side has a mature foundation. NASA's systems engineering guidance treats interactions, integration environments, verification, validation, support, and transition as part of realizing a system; its discussion includes the tooling needed to assemble and test other products [4]. The learning target here is the discovery and improvement of such complete construction processes across new missions.

Computational creativity distinguishes combination, exploration, and transformation of conceptual spaces [5]. DreamCoder learns symbolic abstractions and search guidance, while LAPS examines language-guided learning of program libraries and search heuristics [6, 7]. POWERPLAY searches jointly over new tasks and modifications of a solver [8]. AlphaEvolve provides a concrete precedent for evaluator-guided program improvement in scientific and computational applications [9]. These systems motivate parts of the agenda. Their results do not supply the untested physical, contractual, or economic bridges required here.

Complementarity explains why a coherent set of technical and organizational changes can matter more than isolated upgrades [10]. Endogenous-growth theory supplies a precedent for treating investments in knowledge as part of economic production [11]. Neither precedent makes an individual proposed venture profitable or implies a universal rate of expansion.

2 Complete means: an executable whole

2.1 Purpose first; the method remains editable

Let a construction contract be

$$\mathcal{I} = (\mathcal{Y}, \mathcal{H}, \mathcal{U}, B, T, V), \tag{1}$$

where $\mathcal{Y}$ describes the intended outcome, $\mathcal{H}$ the hard requirements, $\mathcal{U}$ the authority boundary, B the initial resource envelope, T the horizon, and V the acceptance procedure. The environment and

Research object	What it helps explain	Question emphasized here
Systems engineering	Integration and realization of a useful whole.	Can experience improve the construction of a different whole?
Program/library learning	Reusable abstractions and better program search.	Does the gain survive complete realization and equal-content controls?
Transformational intelligence	Changing a restrictive problem-solving setup.	What must actually be built to make the change operational?
Economic invention	Joint capability and voluntary arrangement.	Can the whole sequence be started, delivered, and sustained?
GENESIS	Repeated opportunity discovery and successor effects.	Which part of the builder improved, and what fresh evidence shows it?

Table 1: A connection among established ideas and the three foundations, rather than a claim that their mechanisms are new.

uncertainties are recorded separately. A finite model can stipulate them; a natural application must investigate whether its model is adequate.

The construction may change the method, representations, interfaces, or arrangements allowed by this contract. It cannot retrospectively improve a result by dropping an inconvenient requirement. If a better interpretation of the purpose is proposed, authorization and a new comparison make the change explicit. An acceptance instrument may also be defective. Its repair requires evidence and a common re-evaluation of candidates, not unilateral self-approval.

Definition 2.1 (Complete means, relative to a contract)**.** *Complete means are an executable combination of knowledge, instruments, processes, resources, and arrangements that can be initialized, operated, checked, and sustained sufficiently to deliver the accepted outcome within the declared scope and horizon. Necessary external dependencies, permissions, and remaining obligations are part of that declaration.*

Complete does not mean self-sufficient in every respect. A software instrument may legitimately depend on an operating system, electricity, and a licensed solver. A laboratory may use an external calibration service. The question is whether a material dependency has a supported place in the route, not whether one organization can reproduce the entire world.

2.2 The witness that connects the parts

For a starting state x_0, write a proposed construction witness as

$$w = (\pi, D, \Sigma, A, E, O). \tag{2}$$

Here π is the executable construction and operating policy; D records dependencies and interface meanings; Σ schedules resource use; A specifies required authority and arrangements; E links claims to evidence; and O records continuing obligations. A policy chooses from available observations. It cannot choose a branch using a hidden environmental fact that no allowed observation reveals.

A *prospective witness* says how a modeled route could work. A *realized witness* additionally records the actual execution, required acceptance, and resulting obligations. A forecast of consent belongs to the former. Only a corresponding authorized action belongs to the latter.

Five questions turn a parts list into a construction:

1. **Can it start?** Every initial prerequisite is available or reached by a feasible preceding action. Mutual promises to create each other do not start a process.

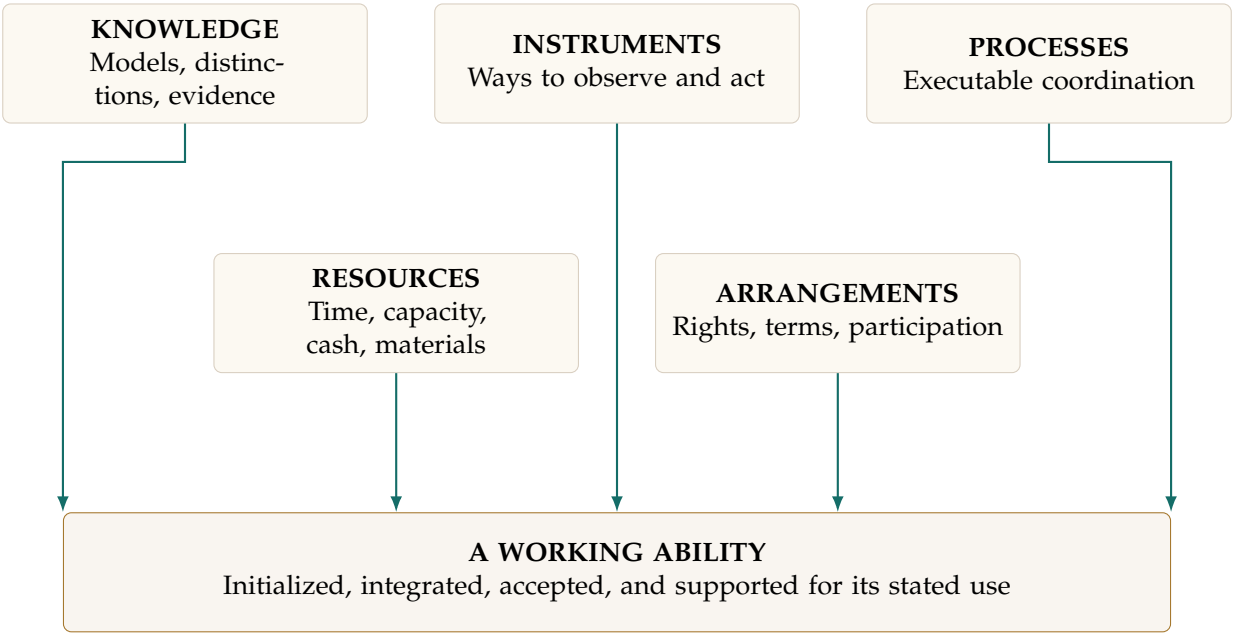


Figure 1: The components enable an outcome together. Lines mean required integration, not independent sufficient causes. A missing or incompatible element can defeat the whole.

2. **Do the parts compose?** Interfaces have compatible meanings, units, versions, and operating conditions. A shared resource is not allocated twice.
3. **Can every step occur?** Time, liquidity, capacity, and authority hold at every prefix of execution, not only in the final totals.
4. **Does the intended outcome follow?** Acceptance checks the useful whole and its adverse cases, including failures shared by several components.
5. **What remains afterward?** Support, maintenance, debts, refunds, decommissioning, and retained capabilities have explicit owners and costs.

These are substantive properties of the route. They need not become five separate committees, scores, or mandatory software modules. A small problem may be settled by one clear calculation and an executable test. A high-consequence physical system requires evidence proportionate to its consequences.

2.3 Completeness has a boundary

For an explicitly modeled environment θ, let $\mathrm{Valid}(w; \mathcal{I}, x_0, \theta)$ mean that the policy's execution respects the contract, its resource prefixes, and its acceptance conditions. Robust model-relative completeness requires this for every θ in a declared set Θ. A probabilistic version requires a stated reliability target under a justified distribution instead.

Neither version establishes that the modeled set contains every relevant real failure. An omitted disturbance, false interface assumption, or forged source can invalidate the conclusion. The witness must therefore distinguish a proved implication, a measured property, an assumption, and an unresolved dependency. Unknown conditions can motivate a new experiment or a construction that makes them irrelevant. They cannot become true because the narrative needs them.

The practical aim is *enough justified completeness for the useful outcome*, with the residual boundary visible. Requiring omniscience would prevent action. Hiding the boundary would manufacture confidence.

3 From available tools to a construction frontier

3.1 A capability is an operation with conditions

Represent the current productive state as

$$x = (\mathcal{K}, \mathcal{J}, \mathcal{P}, r, a, e, o), \tag{3}$$

comprising knowledge, instruments, procedures, resources, authority/arrangements, evidence, and obligations. These categories can overlap in implementation; the notation prevents a useful file from being confused with usable equipment, accepted permission, or spendable funds.

A capability is a conditional operation: given specified inputs and a valid state, it produces an output and a changed state while consuming resources. Construction adds or changes such an operation. Requalification may remove its usable scope. Retirement may reduce obligations. A useful policy must be able to do all three.

For a declared model and horizon, define

$$\mathcal{R}_{\text{use}}(\mathcal{I}, x) = \{y : \text{a valid policy delivers } y \text{ using the installed repertoire}\}, \tag{4}$$

$$\mathcal{R}_{\text{build}}(\mathcal{I}, x) = \{y : \text{a valid policy delivers } y \text{ and may construct new means}\}. \tag{5}$$

The repertoire and the distinction between use and construction must be stated for the comparison. If every use-policy is also allowed in the second set, then $\mathcal{R}_{\text{use}} \subseteq \mathcal{R}_{\text{build}}$. The useful question is whether an accepted outcome lies in the difference under the same initial resources, authority, and horizon.

This is not evidence that an incumbent could never have built the same means. A competent comparator must receive the same construction and procurement options when the claim concerns builder quality. An example can demonstrate a limitation of an installed arrangement while a conventional constructor fully matches the proposed solution.

3.2 Three boundaries that a build can move

The physical or logical boundary. A new observation can distinguish states that old observations cannot distinguish. A new manufacturing operation can change the realizable geometry. Proof requires an appropriate model or independent physical evidence.

The resource boundary. The outcome was possible but not within the available money, time, energy, capacity, or risk allowance. A complete lower-cost route can make it reachable. The claim depends on the stated threshold and a fair complete alternative.

The discovery boundary. An adequate route was feasible, but the procedure did not find it. A better representation, search method, experiment, or construction heuristic can help. This is a gain in finding and realizing possibilities, not necessarily an enlargement of the mathematical feasible set.

The distinctions matter commercially and scientifically. A conventional solution can remove a real obstruction for a particular organization. That does not make the solution new to science. Conversely, a novel algorithm may fail to change any worthwhile outcome after integration costs.

3.3 Complementary prerequisites and the first viable whole

The GENESIS capability hypergraph expresses jointly necessary means [3]. An edge $\{m_1, \ldots, m_k\} \to m'$ is useful only if there is a supported realization, a feasible resource schedule, and conditions under which it remains valid. A hypergraph can summarize the construction; labeling the edge does not construct it.

Proposition 3.1 (An elementary complementarity barrier)**.** *Suppose $|M| \geq 2$, no member of M is initially available, and a useful outcome has value $v > 0$ only when all members are available. Each member has positive construction cost c_i, and no proper subset has other value in this example. A procedure restricted to single-member additions that accepts only nonnegative immediate increments builds nothing. A complete construction is beneficial when $v > \sum_{i \in M} c_i$, provided its path is feasible.*

Proof. Because $|M| \geq 2$, every first single-member addition leaves an incomplete set, produces no outcome value, and incurs positive cost. Starting from no members, the stated local rule therefore refuses every first addition. The complete set has net value $v - \sum_i c_i$. Feasibility is a separate condition because a positive terminal balance need not fund the earlier steps. □

The result is intentionally elementary. It identifies the failure of a particular local rule, not the superiority of a new planner. Ordinary joint optimization can select the same bundle. In practical construction, complementarity suggests testing a small viable whole rather than demanding that each enabling component be independently profitable.

3.4 A loop in the diagram is not a bootstrap sequence

A proposed factory may need a calibration instrument that it is supposed to manufacture. A calibration method may require a reference that the new process is supposed to establish. Circularity is not inherently fatal: an external seed, a temporary method, or a joint operation can start a genuine cycle. The seed and transition must appear in the witness.

Proposition 3.2 (Unseeded prerequisite obstruction)**.** *In a discrete action model, let U be a set of capabilities absent initially. If every permitted action that can create any member of U requires at least one member of U to be present before that action, and no exogenous event supplies one, no finite action sequence creates a member of U.*

Proof. Assume a sequence creates a first member of U. Immediately before that action, no member is present, contradicting the action's required prerequisite. Equivalently, absence of every member is preserved by induction over the sequence. □

This result makes the first construction unusually important. It may require purchased equipment, external expertise, a coarse temporary instrument, a validated reference, or a simpler process. Access to those means must be real at the evidential level claimed. Once initialized, the cycle still needs maintenance, material and energy flows, acceptance, and replacement of its own critical dependencies.

3.5 Building changes possibility; it need not improve it

Let a completed build transform x into $x' = T(x, w)$. The new state includes its outputs *and* its consumed resources, remaining liabilities, and evidence. Comparing $\mathcal{R}(\mathcal{I}, x)$ with $\mathcal{R}(\mathcal{I}, x')$ without accounting for those changes can exaggerate progress. A machine may add useful capacity while tying up all available cash. A library may accelerate one task while making another difficult to maintain.

Proposition 3.3 (Optional capability is weakly valuable under specific conditions)**.** *With the same remaining resources, obligations, environment, and horizon, adding an optional operation cannot reduce optimal attainable value if every old policy remains feasible and the addition imposes no unavoidable cost. Strict improvement requires a newly feasible policy with strictly greater value.*

The proof is feasible-set inclusion. Actual building usually changes resources and can create carrying costs, so the proposition does not say that every construction pays. Nor does a larger feasible set

guarantee that a bounded search procedure finds a better policy. This is why *learning to build* includes learning what to retire, what to buy, and what to avoid building.

4 From a successful build to a better builder

4.1 The construction loop

The proposed builder searches backward from a worthwhile outcome and forward from available means. Backward reasoning asks which missing conditions prevent acceptance. Forward reasoning asks what a new measurement, method, resource, or arrangement makes possible. Their intersection supplies a construction hypothesis rather than a predetermined product catalogue.

The hypothesis must name an obstruction. It can be a proved information limit, a measured capacity shortfall, an incompatible interface, an unfundable cash prefix, an unacceptable allocation of risk, or an uncertain diagnosis. Each calls for a different move. Better optimization cannot supply an absent measurement. A new sensor cannot manufacture consent. More financing cannot repair a physically invalid design.

The builder then compares direct use, acquisition, adaptation, joint construction, an informative experiment, removal of an unnecessary dependency, and non-entry. This is a set of alternatives, not a mandatory sequence. The best candidate becomes an executable witness, is challenged on consequential failures, and is tested against the original outcome. An adjudicated failure should change the construction or its diagnosis. It should not trigger a more favorable success criterion.

> **A useful failure asks for a next construction.**
> "The model was wrong" is a weak lesson. "This calibration cannot distinguish the desired signal from contact drift; obtain an independently varied reference, or change the fixture so drift no longer controls the decision" is a construction-relevant lesson. It remains a hypothesis until its proposed remedy is tested.

Several failures may share a cause. Repeating the same code with different prompts is not independent evidence. Conversely, a critic can be wrong. Retain a checkable counterexample or discriminating observation, adjudicate it, then repair. The loop learns from an established mechanism or a well-described uncertainty, not from the confidence of a critique.

4.2 What should survive the build?

A useful inheritance can have three layers:

1. **The working means:** a program, instrument, interface, procedure, relationship, or other usable asset.
2. **Its applicability:** the purpose it served, assumptions, evidence, failures, dependencies, rights, and conditions that invalidate reuse.
3. **Its constructor:** an executable or reproducible method for creating related means, with its own inputs, costs, and demonstrated scope.

The third layer is often the most consequential and the easiest to overstate. A saved transcript can describe a successful build without being a reliable method for another. A general compiler can be supplied by a developer rather than learned by an agent. A method can work on a new task because the task author encoded the needed adaptation. The evidence must identify which of these occurred.

Selective retention matters. Keep the useful program, a compact mechanism, the essential evidence, and the conditions under which it should not be used. Repeated terminal logs add little when a smaller record preserves the incurred costs, failures, and lesson. Discarding redundant material is compatible with learning; deleting unfavorable results from a comparison is not.

4.3 Four objects that can improve

Let z denote a fresh construction task, K the supplied knowledge and tools, κ a constructor, and B the complete resource budget. Let $W(\kappa, z, K, B)$ report the prespecified accepted outcome together with its relevant costs and failures. When a scalar utility is appropriate, define

$$J(\kappa; K, B, \mathcal{D}) = \mathbb{E}_{z\sim\mathcal{D}}[W(\kappa, z, K, B)]. \tag{6}$$

The task population $\mathcal{D}$ must be meaningful and held fixed for the claim. When benefits are incomparable, report a vector or a Pareto comparison instead of inventing a universal score.

Artifact improvement changes what one produced object does. **Capability inheritance** changes K, perhaps without changing κ. **Constructor improvement** changes κ and is tested with matched K, resources, and task distribution. **Construction-learning improvement** changes a learning rule L that proposes or revises constructors after development experience. It requires fresh campaigns testing the output of L, with development costs included.

Formally, if h is development history and $\kappa' = L(\kappa, h)$, the stronger claim concerns the performance of L across independently initialized development-and-confirmation campaigns. A single useful κ' does not establish that the rule reliably produces better constructors. Searching many variants and reporting only the winner cannot estimate that reliability.

4.4 The finished artifact must sometimes be taken away

An especially informative transfer test removes the donor's finished artifact while preserving the candidate construction procedure. The successor requires a different output, and all arms receive equal permitted task information. If the retained procedure still improves the creation of that output, it has contributed something beyond direct artifact reuse.

This test is necessary for some claims but insufficient on its own. A conventional interpreter, compiler, engineer, or frontier agent may reconstruct the same means as well or better. An *equal-content control* therefore receives the same valid specification, lessons, and construction permissions, while using an ordinary method. A further supplied-solution arm can separate finding the construction from implementing a construction already known.

The experiment in Section 5 deliberately includes the donor-artifact exclusion and an equal-content control. It demonstrates the mechanics of constructing fresh programs. Because the constructor is supplied and the ordinary control ties, it does not demonstrate that experience improved the constructor.

4.5 A better builder sometimes builds less

A retained capability creates options and may create obligations. It can require version updates, inspection, recalibration, scarce expert attention, reserved cash, support, or decommissioning. We call these *construction obligations*; the phrase is descriptive, not a new accounting category. A build is evaluated with them included.

The right decision may be to keep only a generative method, obtain a commodity service when needed, compress several overlapping tools into one simpler representation, or retire an obsolete asset. As

outside methods improve, an internal construction can lose its justification. A discovery enterprise should benefit from that progress rather than preserve its own machinery for prestige.

The retained state is therefore an actively maintained means of future production. Its success is measured by what fresh work becomes feasible and worthwhile, after its costs, rather than by the size of its archive.

5 An executable witness: constructing what the successor needs

The framework should produce something that works. We therefore accompany the paper with a finite study that constructs new executable instruments, checks their outputs, and asks when the complete means of using them are affordable and available. Its purpose is to expose distinctions that a compelling diagram could hide: a finished instrument can fail on a successor; a reusable constructor can build a valid replacement; current rights and qualification remain necessary; and retention can impose obligations that make the next outcome harder to reach.

> **What is supplied and what is executed.** The task language, compiler, specifications, alternative routes, rights scenarios, and planning costs are authored. The generated programs and their output checks are executed. The constructor is *supplied, not learned*. No physical sensor, customer, transaction, model call, or autonomous discovery is represented.

5.1 Constructing a fresh instrument

A task specifies named integer input fields f_j, affine calibration coefficients a_j, b_j, aggregation weights w_j, and a threshold θ. Its required measurement and decision are

$$z(x) = \sum_{j=1}^{d} w_j \big(a_j x_{f_j} + b_j\big), \qquad d(x) = \mathbf{1}\{z(x) \geq \theta\}. \tag{7}$$

The compiler combines these terms into a standalone Python program. It consumes JSON records, requires the exact integer schema, computes the measurement and Boolean decision, and emits a specification hash. This is actual source generation from a supplied specification, using ordinary algebra. It is not a discovery of a new measurement law.

The source instrument computes $2a + 2b + c + 1$. The first successor instead requires $2q - p + 6r + 1$, with different field names, coefficients, and threshold. Further successors require four and five fields. The finished source program cannot consume these schemas. Keeping that program alone therefore does not solve the successors. The retained generic compiler creates three fresh programs from their specifications; their correctness does not depend on executing the finished donor program.

For each task we enumerate every vector in $\{-2, -1, 0, 1, 2\}^d$. A separate calculation checks both the numerical result and the decision. A strong ordinary control directly interprets the same declarative specification, without needing to generate a program. Table 2 reports all cases. The output field named `accepted` denotes the per-record threshold decision; commercial acceptance is not implied. Program qualification means matching the specified result on the entire declared test domain, including correctly negative decisions.

Task	Input cases	Both methods correct	Finished source program
Source: three fields	125	125	Same task
Successor 1: three fields	125	125	Schema rejection
Successor 2: four fields	625	625	Schema rejection
Successor 3: five fields	3,125	3,125	Schema rejection
Total	**4,000**	**4,000**	Three incompatible successors

Table 2: Actual generated programs and the ordinary interpreter agree with the specifications on the complete stated finite domain. These authored fresh tasks are not independently selected natural tasks. No runtime or learning advantage is claimed.

5.2 Completing the means of delivery

Correct output is one component of the modeled outcome. An internal delivery also requires a reusable testing fixture, current access terms, fresh qualification, and production resources. An external complete service is a separately available conventional route under its own supplied standard terms. Missing internal consent does not disable that external alternative.

The fresh internal route costs 54 planning units and takes eight time units: 18 for construction, 10 for the fixture, 6 for current terms, 8 for qualification, and 12 for production. Inheriting the fixture removes its repeated cost and time, giving 44 units and six time units. Inheriting the supplied constructor additionally changes assumed fresh construction cost from 18 to 7 and its duration from three to one, giving 33 units and four time units. The complete external service costs 58 units and takes four time units. These are *synthetic planning assumptions*, not measured savings from compilation, labor estimates, currency, or market quotations.

The constructor state also carries a current maintenance obligation m. This must be paid even if the external service or nonentry is selected. Every delivery expense precedes its modeled receipt of 80 units. A route is feasible only if every cash prefix remains nonnegative, the deadline is met, and its current rights and qualification conditions hold. Constructor acquisition and full continuation costs are accounted for separately in the economic analysis; the state comparison here conditions on what has already been retained.

5.3 The frontier expands—and can contract

Before execution, the protocol fixed cash values $\{32, 33, 43, 44, 53, 54, 57, 58\}$, deadlines $\{3, 4, 5, 6, 7, 8\}$, two internal-consent settings, and maintenance values $\{0, 12, 26\}$. Their Cartesian product contains 288 cells. Evaluating the three inherited states gives 864 complete-route decisions. Each decision selects the least-cost feasible supplied route, or nonentry when none qualifies. This is exact only within the declared catalogue.

The positive witness is transparent. With cash 33, deadline four, current internal consent, fresh qualification, and zero maintenance, the constructor route delivers. The fresh, fixture-only, and external routes do not fit those resources. Remove current rights or qualification and the internal result becomes infeasible.

The adverse witness is equally consequential. With cash 58, deadline four, and no internal consent, the fixture-only state can buy the external complete service. Retaining a constructor with maintenance 12 leaves insufficient cash for that same service. The additional asset has contracted the reachable set. This does not contradict conditional state dominance: its no-greater-obligations premise has failed. A useful retained object is not automatically a better productive state.

Maintenance m	Cells	Fresh state	Fixture-only	Constructor state
0	96	12	22	**40**
12	96	12	**22**	20
26	96	12	**22**	0

Table 3: Number of cells admitting an accepted modeled delivery. The selected grid is a finite comparison, not a probability distribution. Fresh and fixture-only counts repeat because the maintenance obligation belongs only to the constructor state. These unequal-state counts are not a ranking of learning algorithms.

5.4 What the strong control establishes

An ordinary analytical planner receives the same specifications, constructor recipe, retained fixture, resource limits, rights, costs, and complete routes. It agrees with the constructor-state planner in **all 288 decisions**. Its ordinary interpreter also matches every generated instrument's required output. The control is therefore not deprived of the very content responsible for the conditional gain. Compilation is not technically necessary for these arithmetic tasks: the interpreter already supplies the required answers. The emitted programs demonstrate portable construction, not the uniquely necessary or empirically cheapest implementation.

The study establishes a finite example of *fresh construction using retained means*. It does not establish that experience improved the constructor, that a special discovery policy outperformed ordinary engineering, or that the supplied cost advantage would occur in practice. The conventional tie is a substantive result: productive value and proprietary methodological superiority are different claims. The three fresh schemas demonstrate donor-artifact incompatibility; they do not establish open-ended transfer outside the supplied language.

5.5 Reproduction and claim boundaries

From `docs/learning-to-build/study`, run:

```
python3 study.py --output evidence
python3 check.py --evidence evidence
```

The first command generates the instruments, compact results, and all grid cells. The second imports no producer code: it separately executes the instruments and replays route costs and cash prefixes. This is a separate-source engineering cross-check, not an independent institutional evaluation. The retained protocol and result hashes bind the task definitions and implementation. Both entry points pin the exact registered protocol; this reproduction is not a general configurable simulator. The checker recomputes the complete reported record, including all counts, hashes, flags, and row cardinalities. Normal and optimized Python execution produced byte-identical reported results.

Adverse checks reject incompatible schemas, a tampered task contract, missing current qualification, and absent internal rights, while preserving the maintenance counterexample. Exact checking covers the authored domain and assumptions. It does not establish real demand, observed consent, physical qualification, measured economic productivity, or construction learning. The next experiment must earn those stronger bridges rather than rename this witness.

The lesson. Keeping a finished object and keeping a means of building the next object are different achievements. Either becomes productive only when the complete current means of accepted use are available.

6 Buildability includes the means of paying for the build

6.1 The arrangement can be the missing instrument

A technically feasible construction may remain unused because its beneficiaries are fragmented, its risks are misallocated, or its funding arrives too late. A shared testing facility, a staged acceptance contract, compatible access rights, or a division of responsibilities can be part of the complete means. Such arrangements are construction hypotheses, not inferred agreements.

For participant i, a proposed arrangement should compare its expected utility U_i with an explicit outside option U_i^{out}. Modeled voluntary participation requires

$$U_i \geq U_i^{\text{out}} \qquad \text{for each required participant } i. \tag{8}$$

This condition is neither observed consent nor a proof of truthful behavior or coalition stability. It depends on supplied beliefs, information, alternatives, and enforceability. Real acceptance must be obtained at the appropriate stage. New technology can improve competitors' outside options as well as the proposed builder's costs.

Invent the Economy makes this joint search explicit [2]. A component with poor stand-alone returns can belong to a worthwhile package. Conversely, positive combined surplus may fail to support acceptable individual terms. The construction brief should identify who benefits, who contributes, when each acts, what each can refuse, and which complete alternative each could choose.

6.2 Every cash prefix must be feasible

For a declared cash schedule, write cumulative paid construction and operating costs as $C(t)$, cleared receipts as $R(t)$, received external financing as $F(t)$, financing repayments and distributions as $D(t)$, and required protected funds as $Q(t)$. With opening cash b, available cash is

$$A(t) = b + R(t) + F(t) - C(t) - D(t) - Q(t). \tag{9}$$

The route requires $A(t) \geq 0$ at every event time. $Q(t)$ must exclude amounts already counted as paid costs or repayments; releases from protection reduce Q rather than create revenue. The equation is a cash-feasibility statement, not a replacement for an operating ledger or an accounting standard.

For a fixed schedule without new financing, the minimum opening cash implied by these events is

$$b_{\min} = \max\left\{0, \sup_{t \leq T}\left[C(t) + D(t) + Q(t) - R(t)\right]\right\}. \tag{10}$$

A positive terminal contribution does not remove an earlier funding shortfall. A promised investment is not received financing. Refund exposure, supplier commitments, and support obligations can remain after collection. A lower complete cost is useful only when the revised sequence can actually occur.

Starting capital is a design variable for a new economic comparison. Small staged construction, outside financing, a larger initial facility, and non-entry may all be legitimate alternatives. Their terms, costs, downside, and timing belong in the comparison. An earlier fixed-capital experiment retains its original assumptions; it is not silently topped up to make a new result attractive.

6.3 What the finite study says about continuation

The study uses synthetic planning units, with no currency interpretation. Its source route costs 54 and produces an assumed receipt of 80 after accepted delivery. Preserving the fixture alone leaves a successor cost of 44. Acquiring the supplied reusable constructor costs 9; each successor then costs $33 + m$, where m is its current maintenance obligation. Thus, for n successors,

$$C_{\text{fixture}}(n) = 54 + 44n, \tag{11}$$

$$C_{\text{constructor}}(n, m) = 54 + 9 + n(33 + m), \tag{12}$$

$$\Delta C(n, m) = n(11 - m) - 9. \tag{13}$$

Constructor retention is strictly cheaper exactly when $n(11 - m) > 9$, conditional on the same acceptance, valid current rights, deadlines, fixture availability, and feasible cash prefixes. At $m = 0$, three successors save 24 units after acquisition. At $m = 12$, every additional successor makes this cost comparison worse. The parameters illustrate a threshold; they are not observed market costs or evidence of increasing returns in a real enterprise.

The distinction between *later*, *financed by*, and *enabled by* is visible even here. An illustrative cash path starting at 54 spends the source cost, collects the modeled 80, pays 9 for acquisition, and can prepay the next 33-unit delivery. Yet the source's net contribution is only 26. If all opening capital must remain protected, the 26 does not cover acquisition plus the next 33. The path is feasible using available modeled cash; it is not, on that rule, fully funded by unrestricted earnings alone.

Likewise, a retained constructor can make a 33-unit successor affordable where a 44-unit fixture-only route is not. To establish this as a *causal resource advantage*, compare equally resourced successors that may build or buy alternative means. To establish the value of acquiring the constructor, charge its acquisition and prior failed development as well. State-level reachability and whole-lineage investment value answer different questions.

6.4 Value creation and value capture must remain connected

An enterprise can use accepted value to sustain future construction through prices, subscriptions, licensing, shared facilities, or other voluntary arrangements. The form should reward the useful outcome and bear the necessary obligations. Billing unnecessary operations can conflict with an intelligence whose best contribution is to eliminate those operations.

Complete economic cost includes search, unsuccessful candidates, development, acquisition, production, checking, integration, maintenance, support, financing, and residual liabilities. The costing policy should explain amortization and allocation without counting one expense twice or crediting the same receipt to several generations. Included assistant resources do not imply zero underlying economic cost; unavailable bills remain unknown.

Commercial recurrence need not prove scientific novelty. A known method can support an excellent business. Scientific construction need not immediately produce a sale. The desired flywheel joins useful capability, voluntary accepted delivery, resources available for continuation, and a tested successor contribution; each connection retains its own evidence.

7 How to establish that intelligence learned to build

7.1 A prospective campaign, not a gallery of successes

The empirical target is a sequence of consequential construction tasks whose acceptance is defined before results are observed. Development tasks may reveal useful mechanisms. Confirmation tasks must require fresh outputs, come from an independently motivated population, and be protected from repeated adaptation. Many inputs to one generated program are correctness cases, not independent inventions.

A practical campaign begins with a valuable outcome that current complete alternatives do not adequately serve under a stated constraint. Compare at least three materially different possibilities when selecting a new flagship, rather than several cosmetic variations of a convenient benchmark. Selection evidence should explain the potential consequence and the decisive first construction, including why the strongest available complete alternative leaves a residual.

The selected campaign records every registered attempt, complete resource use, accepted outcomes, and compact failure lessons. It permits negative results, ordinary winners, justified refusal, and retirement. It does not restart the same failed comparison under a new title until a favorable result appears.

7.2 Cross the method with the means

At fixed underlying model, information access, and budget, cross the old and new constructors with the old and new capability stock:

	Original means K_0	**Retained means** K_1
Original constructor κ_0	Baseline construction.	Effect of additional content and tools under the old method.
Revised constructor κ_1	Procedure effect without the new asset stock.	Joint effect and interaction between procedure and stock.

Table 4: A causal comparison structure, not a claim that the four cells have been measured in this edition. Starting costs and obligations must be aligned for the intended estimand.

For a common held-out population, define the procedure effect at fixed means as

$$\Delta_\kappa(K_j) = J(\kappa_1; K_j, B, \mathcal{D}) - J(\kappa_0; K_j, B, \mathcal{D}). \tag{14}$$

The interaction is the difference between these effects at K_1 and K_0. A revised method may be useful only with the representation it helped create. That is a joint mechanism, not a reason to hide the other cells. Cross underlying model versions as well when attributing gains to institutional learning rather than external model progress.

There are two valid resource questions. A *current-state* test asks what can be built now from a specified inherited state. A *lineage* test asks whether acquiring and maintaining that state improved total results. The latter charges development, failed search, migration, and maintenance to the appropriate arms. Neither should quietly answer the other's question.

7.3 Controls that make the claim interpretable

Artifact exclusion. Withhold the donor's finished output; require a genuinely different successor. Retain the proposed construction method and its legitimate information.

Equal-content ordinary reuse. Supply the same specification, lessons, data, and means to a competent conventional method. This tests whether the proposed procedure adds value beyond saved content.

Strong unrestricted construction. Allow a capable specialist, frontier agent, or engineer the same construction, procurement, coordination, and financing permissions. A control restricted to old tools cannot establish superiority over the best complete route.

Supplied-solution realization. Give each arm a correct construction description. If the advantage disappears, the difference concerned finding the construction; if it persists, investigate implementation, integration, or verification rather than calling everything discovery.

Dependency and obligation perturbations. Change interfaces, revoke only the permissions actually being varied, invalidate calibration, remove the fixture, introduce capacity conflicts, and test maintenance. Retained capability is conditional on these facts.

Negative opportunities. Include cases where the old method is already adequate, construction does not pay, or no valid route is found. Measure missed useful opportunities and invalid promotions as well as accepted successes.

These controls can be selected for a narrower registered claim; every project need not run every possible arm. Omitted controls limit attribution. They do not justify silently asserting the stronger conclusion.

7.4 Measure outcomes, not the appearance of activity

The primary engineering outcomes are acceptance, complete cost and time to acceptance, resource feasibility, and the scope of the constructed capability. Record interventions, corrections, failures, support burden, and whether the successor depends on retained means. Report distributions or paired outcomes when there are independent tasks, with uncertainty appropriate to the sampling and stopping scheme.

When measuring productivity, distinguish a competent frontier-AI operation from competent human workers using conventional software under the same useful outcome. Count actual or explicitly modeled human intervention inside the AI operation. Preserve ordinary software's genuine automation in the human comparator. Human-plus-AI providers, other autonomous providers, and self-service are additional commercial alternatives. None may be weakened to manufacture a result.

A learned quality estimate cannot substitute for the actual acceptance boundary. Multiple AI reviewers can share a blind spot. A separately implemented checker is useful procedural separation, not an external institutional audit. Natural scientific and industrial tasks need independent measurements, domain-valid uncertainty, and evidence about the observations the evaluation itself might miss.

7.5 A decisive next application

The motivating application is a *construction assistant for scientific instruments*: given an authorized measurement purpose, produce a usable acquisition or analysis instrument, its interfaces and calibration procedure, and evidence that it supports the intended decision. The product hypothesis is an ability the user can operate and maintain, not a transcript of agents discussing a build.

An initial physical campaign could ask whether a new observation or fixture separates an internal sample property from contact effects that defeat the installed setup. The first test must compare strong existing

inverse methods, uncertainty analysis, and equally permitted engineering. If an existing complete route already answers the question, adopt it and revise the alleged missing means. A simulator cannot validate an observation never acquired.

After the first qualified construction, freeze the retained method before revealing a different measurement objective. Compare fresh construction with and without the method, including an equal-content conventional route and all adaptation costs. Only then test a voluntary delivery arrangement. This sequence addresses a consequential dependency of the GENESIS programme without treating the present paper as a completed instrument business.

8 The horizon: build the means of further construction

8.1 Scientific instruments are a particularly valuable beginning

A new instrument can change which questions are answerable. A better experiment can change which design uncertainties remain binding. A production process can then turn that knowledge into repeatable material capability. This gives a construction a second possible consequence: it may improve the means of discovering and realizing another construction.

The relationship is causal only when the successor actually depends on the new means relative to strong alternatives. A later project that uses the same codebase is not enough. Neither is a story that every useful invention somehow contributes to everything after it. Identify the specific observation, operation, representation, resource, or arrangement the predecessor supplied; remove or replace it in a fair comparison; measure the successor consequence.

Three ambitious candidate families illustrate different mechanisms. A new *measurement operation* can expose an otherwise unidentifiable property. A new *conversion or manufacturing process* can turn an unusable input into a qualified component. A new *productive arrangement* can make complementary facilities, demand, expertise, and funding viable together. These are research directions, not completed products of this edition.

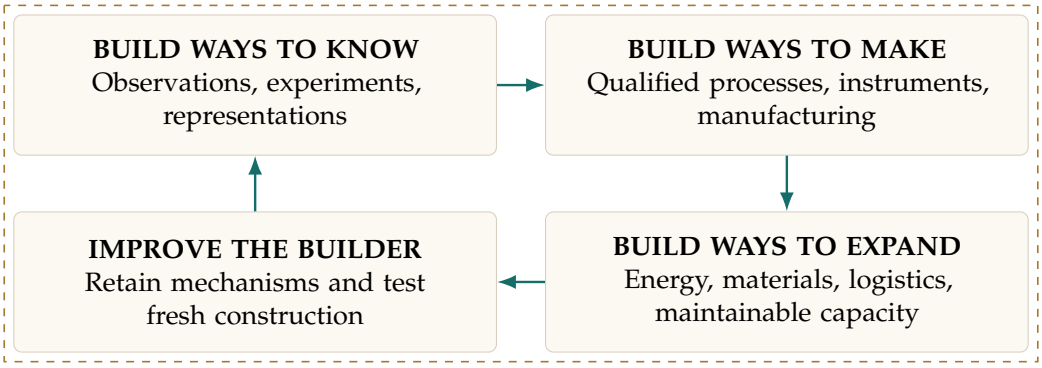


Figure 2: The long-horizon hypothesis. Each arrow requires its own accepted construction and supported enabling relationship. The loop does not imply a fixed sequence or guaranteed growth.

8.2 Knowledge can travel; capacity has to be made

A useful design can be reused without rebuilding the original reasoning. This is an important property of knowledge, connected to the economic distinction between rival resources and nonrival technological knowledge [11]. The design's physical realization still requires materials, energy, equipment, fabrication, qualification, logistics, and time. Its rights and applicability may also limit transfer.

Consider a prospective production batch u_t, a vector of proposed activities. Necessary resource conditions can be written

$$A_t u_t \leq r_t, \qquad e_t^{\mathsf{T}} u_t \leq E_t, \qquad h_t^{\mathsf{T}} u_t \leq H_t. \tag{15}$$

Here A_t records material and other capacity requirements in declared compatible units, r_t their available amounts, E_t the energy budget over the batch interval, and H_t the admissible heat-rejection budget over that interval. The coefficient vectors record the corresponding batch requirements. Scheduling, peak power, environmental conditions, quality, losses, and maintenance add further constraints; these aggregate inequalities alone are not sufficient for operation.

Intelligence can change the coefficients by inventing a different process. It can find an alternative material, reduce wasted work, improve measurement, or remove an unnecessary dependency. It cannot simply omit a term because the result would otherwise be unaffordable. A cash balance is not an energy supply, and a bill of materials is not a commissioned facility.

The ambitious target is productive infrastructure that helps construct more productive infrastructure. Qualified sensing supports process control; control supports fabrication; fabrication supports additional instruments and energy systems. Replacement parts and repair are part of this loop. A system that expands quickly while consuming an unreplaceable critical input is not self-sustaining under that boundary.

8.3 Beyond one facility, beyond one planet

The horizon extends to extensive solar-powered production, autonomous construction, and scientific infrastructure beyond Earth. This is not an untouched conceptual territory. NASA's current lunar technology programme explicitly combines power, resource processing, manufacturing, construction, and environmental challenges; its description includes integrated tests using regolith simulants [12]. Such existing complete routes belong among the strongest alternatives, not among accomplishments to relabel as ours.

The research question is which remaining constraint prevents a particular useful system from operating, and what new construction can remove it. If the constraint is a missing reference or unreliable conversion step, a precise small experiment may be more consequential than a broad architectural demonstration. If it is logistics or financing, another algorithm may be beside the point. If the incumbent already meets the purpose, the discovery programme must look for a different residual rather than invent one.

Expansion to much larger energy and material throughput would require many separately qualified advances: long-lived power systems, resource acquisition, manufacturing chains, repair, transport, thermal management, coordination, and legitimate governance. No measured result here supplies an expansion rate, deployment date, or wealth forecast. The paper's contribution is to make the route a sequence of buildable, testable dependencies rather than a leap from intelligence to an assumed outcome.

The durable ambition is an increasing ability to create useful abilities.
Scientific knowledge matters through what it lets us understand and do. Productive wealth matters through worthwhile possibilities it sustains. A successful construction programme connects them by building, checking, maintaining, and improving the means.

8.4 Expansion should enlarge useful choice

Private returns can fund research and construction, but they are not the same quantity as total benefit. Competition and diffusion can reduce one firm's margin while making a capability useful to more people. A worthwhile system must remain compatible with voluntary participation, honest risk allocation, exit, and meaningful alternatives.

Construction can also create harm, concentrate control, exhaust resources, or make a community dependent on fragile infrastructure. These are consequences to investigate and govern, not external details to ignore when counting productive capacity. The relevant objective is sustained useful possibility under legitimate authority, with affected parties and material obligations represented.

There is no universal business-size ceiling in the framework. Nor is there a guarantee that bigger is better. Real resource constraints, end-use value, maintenance, coordination, environmental effects, and alternatives determine the useful next scale.

9 Limits, falsifiers, and the next result worth earning

The finite study supplies its task language, calibration coefficients, costs, times, rights flags, and construction method. It therefore tests implementation and conditional reachability, not the autonomous discovery of those objects. Its three successor tasks vary within one authored arithmetic family. Four thousand input checks do not amount to four thousand independent scientific discoveries. The conventional control ties; the study provides no measured constructor-learning advantage.

The formal results are elementary consequences of declared transition, value, and resource assumptions. Real completeness depends on the adequacy of the declared boundary. Open environments can introduce new failures, participants can decline or change terms, and source evidence can be wrong. A finite exact optimum is an optimum only in its supplied feasible set.

The ambitious thesis would be weakened if retained construction methods failed on independent tasks, if ordinary equal-content routes explained their entire apparent advantage, if maintenance consumed the savings, or if useful outcomes could not support a workable arrangement. These are productive possible conclusions. An enterprise can still adopt a useful conventional tool without attributing to itself an improved invention process.

The next result worth earning is one complete consequential construction with independent acceptance, followed by a fresh construction whose improvement is causally attributable to what was retained. An economical accepted delivery would establish a further bridge. Repeated improvement of the constructor across independently initialized campaigns would establish a stronger one. These are concrete tests of the vision.

10 Conclusion: the possibility after the possibility

Intelligence becomes materially consequential when it helps make worthwhile outcomes attainable. That often requires more than solving a problem inside an inherited setup. It requires discovering the missing means, constructing a coherent whole, and learning which parts of the experience make new construction better.

The complete means may be a measurement, a representation, a process, a material capability, a resource, or a voluntary arrangement. The best construction can combine them, replace them, or make some unnecessary. What survives should improve a later ability, with its applicability and obligations visible.

The programme is ambitious because the object of improvement is the means of making progress itself. It is testable because each step names a useful outcome, a construction, a complete alternative, and evidence that the change mattered.

Change the game.
Build what the new game requires.
Learn to build better.

When we build something new, we change what can be built next.
The task is to make that change worthwhile—and learn how to do it again.

A A one-page construction brief

Use this brief to turn an attractive possibility into a concrete construction decision. Answers can be short; unresolved items remain explicitly unresolved.

Question	What to write
Worthwhile outcome	Who should be able to do what? State acceptance, deadline, essential quality, and the material consequences of failure.
Best complete route	What can a competent existing provider, engineer, or frontier agent already deliver under the same permissions and resources? Include self-service and non-entry.
Missing means	Which precise observation, instrument, representation, process, resource, interface, or arrangement prevents the outcome? Is this proved, measured, or only suspected?
Smallest decisive build	What executable artifact, experiment, or arrangement would test that diagnosis and connect back to the useful outcome? What finding would refute it?
Bootstrap and integration	Where do the initial prerequisites come from? Which interfaces, shared resources, environment conditions, and temporary means must work together?
Complete resource path	Count discovery, failed attempts, construction, checking, production, transition, maintenance, financing, and remaining obligations. Verify each prefix, not only the final total.
Participation and authority	Who must agree, what can they choose instead, and what evidence establishes their actual permission? Which conditions are only modeled?
Useful inheritance	What working means, validity conditions, and construction procedure should survive? What should be retired?
Fresh successor test	Name a different useful output. Remove the finished donor artifact where appropriate. Compare equal-content construction and charge adaptation and requalification.
Decision and learning	Build, adopt, test, defer, or decline. Record tried $\rightarrow$ outcome $\rightarrow$ reason or uncertainty $\rightarrow$ next use or avoidance.

The brief is useful only when it changes a construction or operating decision. It is not an admission badge and does not establish demand, physical feasibility, or learning by itself.

B Formal details and interpretation

B.1 A qualified dominance statement

Let $\mathcal{F}(x)$ be the accepted outcomes reachable in a finite transition model from state x. Suppose state x_B preserves every x_A-feasible action sequence and its acceptance and authority conditions; supplies at least the required resources at every corresponding prefix; has no greater relevant obligations; and permits completion no later. Then

$$\mathcal{F}(x_A) \subseteq \mathcal{F}(x_B). \tag{16}$$

Replay any valid x_A sequence in x_B. Each transition remains permitted and feasible by the assumptions, so its accepted outcome remains reachable. The conclusion is a simulation argument over the stated transition systems. Adding an asset does not, by itself, establish the assumptions. Maintenance-driven contraction in the finite study violates them.

B.2 Why generic diminishing returns need not hold

Let i be an instrument and q its necessary current qualification, with $V(S) = 1$ exactly when $\{i, q\} \subseteq S$, and $V(S) = 0$ otherwise. Then

$$V(\{q\}) - V(\varnothing) = 0 < V(\{i, q\}) - V(\{i\}) = 1. \tag{17}$$

The marginal contribution of qualification is larger when the instrument is present. The usual submodular diminishing-returns inequality therefore fails. A generic guarantee for a submodular selection algorithm cannot be borrowed without checking its assumptions. The example supports joint reasoning about prerequisites; it supplies no algorithmic advantage over ordinary joint optimization.

B.3 Three cost boundaries for a successor claim

Current-state boundary: the inherited assets and obligations are specified inputs; the test asks what the successor can do now. **Acquisition boundary:** the decision to obtain the assets is included, with its cost and time. **Whole-lineage boundary:** source construction, failed development, preservation, all successors, and residual obligations enter the accounting.

All three can be informative. In the study, the 33-unit affordability witness is a current-state result, while Equation 13 includes the 9-unit acquisition cost and common source cost. Neither is a measurement of the research effort required to invent the supplied compiler. Switching boundaries after seeing results would change the question.

C Reproduction and source lineage

The publication includes editable LaTeX, a bibliography, the frozen study specification, generated standalone programs, compact results, and a separately implemented checker. From the paper directory, the core numerical reproduction is:

```
python3 study/study.py --output /absolute/new/replay
python3 study/check.py --evidence /absolute/new/replay
python3 build.py
```

The source archive provides a short start guide and the same commands. Study computations use Python's standard library. Typesetting uses `pdflatex`, BibTeX, and the packages listed in the source, with shell escape disabled. The canonical PDF's SHA-256 identifies the published bytes. Recompilation on a different TeX distribution may change bytes while preserving content; byte equality is checked only within a common toolchain.

Protocol status. The supplied finite protocol was written and locally frozen before its first generated-program run in this preparation workflow. That is an author-controlled registration, not an externally timestamped preregistration or independent task selection. All registered grid cells are retained. Reproduction binds the protocol and the resulting claims to the actual checked source.

Source editions. The three supplied foundations are byte-identical to the public repository editions inspected at revision `22678d629d986572c50891b60455911978789da5`. Their exact SHA-256 identities

and section/page map are recorded in `SOURCE_MAP.md`. For the GENESIS PDF, the physical page index is one greater than its printed page after the cover. No source upload or private conversation transcript is included in the publication.

Review. AI-assisted drafting, coding, literature checks, and separate adversarial review were used under the author's direction. Review corrected a missing multi-prerequisite condition in the complementarity statement and strengthened the study's binding between declared inputs, recomputed results, and reported claims. This is editorial and engineering quality control, not external peer review, independent institutional certification, or an empirical benchmark of frontier models.

D Claim and evidence map

Claim	Evidence in this edition	Boundary
Complete means are a useful unit of construction	Definition, witness, interfaces, bootstrap, prefix resources, and acceptance.	Operational synthesis; no exhaustive world model.
Complementarity can defeat local additions	Proposition 3.1 and explicit assumptions.	Particular local rule; ordinary joint planning can succeed.
Unseeded cycles cannot initialize	Proposition 3.2.	Stated action model; exogenous supply or a new operation can change it.
Fresh executable instruments are constructed	Four emitted Python programs; 4,000 finite input cases; independent interpretation.	Authored arithmetic tasks; supplied compiler; no physical sensing.
Retained means can expand or contract modeled reachability	288 supplied cells, 864 decisions, favorable and maintenance-adverse slices.	Assumed costs, timing, consent, and acceptance; not probabilities.
Conventional equal-content method matches	288/288 route ties and equal finite program answers.	No measured proprietary method or learning advantage.
Learning improves construction	Prospective artifact-excluded and factorial tests.	Unestablished in this edition.
Accepted commerce funds a successor	Cash conditions and illustrative continuation arithmetic.	No real customer receipts or native enterprise operation here.
Construction can support much larger productive capacity	Dependency-based scientific and industrial research programme.	No achieved physical scale, expansion rate, or wealth forecast.

References

[1] Vincent Boucher. Transformational intelligence: Learning to change the game, October 2026. URL https://montrealai.github.io/AGI-Alpha-Invention-Flywheel-Lab/transformational-intelligence.html. Research manuscript.

[2] Vincent Boucher. Invent the economy: Transformational intelligence for creating viable markets, October 2026. URL https://montrealai.github.io/AGI-Alpha-Invention-Flywheel-Lab/invent-the-economy.html. Research manuscript.

[3] Vincent Boucher. ALPHA GENESIS FLYWHEEL Ω: Riding the exponential—three orders of intelligence for an enterprise that discovers what becomes possible, November 2026. URL https://montrealai.github.io/AGI-Alpha-Invention-Flywheel-Lab/alpha-genesis-flywheel-omega-v2.html. Research edition 2.0, 4 November.

[4] NASA. NASA systems engineering handbook. Technical Report NASA/SP-2016-6105 Rev2, National Aeronautics and Space Administration, 2016. URL https://www.nasa.gov/reference/systems-engineering-handbook/. Chapter 5: Product realization; especially product integration and qualification of reused products.

[5] Margaret A. Boden. Creativity and artificial intelligence. *Artificial Intelligence*, 103(1–2):347–356, 1998. doi: 10.1016/S0004-3702(98)00055-1.

[6] Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B. Tenenbaum. DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning, 2020. URL https://arxiv.org/abs/2006.08381.

[7] Lionel Wong, Kevin M. Ellis, Joshua Tenenbaum, and Jacob Andreas. Leveraging language to learn program abstractions and search heuristics. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11193–11204, 2021. URL https://proceedings.mlr.press/v139/wong21a.html.

[8] Jürgen Schmidhuber. POWERPLAY: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem, 2011. URL https://arxiv.org/abs/1112.5309.

[9] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

[10] Paul Milgrom and John Roberts. The economics of modern manufacturing: Technology, strategy, and organization. *American Economic Review*, 80(3):511–528, 1990. URL https://www.gsb.stanford.edu/faculty-research/publications/economics-modern-manufacturing-technology-strategy-organization.

[11] Paul M. Romer. Endogenous technological change. *Journal of Political Economy*, 98(5, Part 2):S71–S102, 1990. doi: 10.1086/261725. URL https://www.nber.org/papers/w3210. Open working-paper version: NBER Working Paper 3210 (1989).

[12] NASA. Lunar surface technology, 2026. URL https://www.nasa.gov/lunar-surface-technology/. Official programme description, accessed 7 November 2026. Statements concern described projects and tests, not a deployed self-reproducing industry.