00 **INTRODUCTION** | Reading guide and investigation path

09 · READING GUIDE

AGI X

CURRENT AGI APPROACHES × TMC PROPOSAL

> **“Before we scale toward artificial general intelligence, we must make explicit what kind of intelligence we seek — and how we will recognize its realization.”**

WHAT IT IS

The chapter compares contemporary AI and AGI approaches through a common question: **what does each technique transform in the system that will face the next problems?**

With AGI X, the TMC applies Factor X to investigate the target of construction. It proposes an architecture in which the effects of learning, retrieving, evaluating, and acting take part in the historical transformation of operational criteria.

KEY INSIGHTS

- **Solving a new problem and transforming the system that will solve subsequent ones are distinct achievements.** What matters is to investigate what changes, how long it persists, and which future operations that change conditions.

- **Integrated capabilities can transform one another.** The TMC investigates a common trajectory in which learning, memory, abstraction, evaluation, and action modify one another’s conditions of operation.

WHAT FOR

To compare paths toward AGI, identify convergences and differences with the TMC, and derive requirements for an architecture capable of historical transformation. This examination prepares the engineering contract for the Apparatus in Chapter 10.

WHAT IS ITS EFFECT

In addition to asking:

> **"How many tasks can the system solve?"**

we begin to investigate:

> **"How does solving those tasks transform the system that will face the next ones?"**

In addition to asking:

> **"How can information be stored, retrieved, and processed?"**

we investigate:

> **"How do the effects of these operations modify the criteria for selection, retrieval, learning, and action?"**

Existing techniques already realize parts of this movement and exhibit different degrees of integration. The TMC proposal examines how to articulate them along a common trajectory inspired by the cognitive organization of living beings.

FOR WHOM / CONNECTIONS

- **AI / AGI** — generalization, continual learning, agents, world models, and cognitive architectures.

- **Computing / engineering** — State, memory, processing policies, and persistent effects of execution.

- **Neuroscience** — plasticity, metaplasticity, and transformation of the conditions of learning.

- **Biology / living systems** — continuity, adaptation, and organization under resource constraints.

- **Cognitive science** — relationships between the acquisition of capabilities and the transformation of the learner.

- **Chapter 8 — Cognition and Feeling** — abstraction, selection, and efficiency.

- **Chapter 10 — AGI: Programming** — formalization of requirements and computational organization of the Apparatus.

- **Chapter 11 — Demarcation of Life** — distinctions among intelligence, adaptation, sentience, and life.

> **ROLE IN THE THESIS**

To initiate Movement IV — **TRANSLATE AND TEST** — by confronting the TMC with AGI problems and solutions. Chapter 9 organizes the requirements; Chapter 10 investigates their formal and computational translation, without treating that translation as a demonstration of mind, sentience, or life.

PATH THROUGH THE INVESTIGATION

01 AGI X · FIVE POSTURES
02 IMPASSE
03 PARADOXES
04 SCALE AND CREATION
05 MEMORY AND METHOD
06 AGI AND LLM
07 GENERALITY
08 LEARN AND ACT
09 CAUSE AND SEARCH
10 COMPONENTS AND ECOSYSTEM
11 THE AGI DILEMMA
12 APPARATUS

9.0.1–9.0.3

AGI X · THE FIVE POSTURES

The map of the proposal and its technical limits

IN THIS BLOCK

9.0 POSTURES | THE AGI X PROPOSAL MAP
9.0.1 AGI × AGI X: WHAT ARE WE PROPOSING TO BUILD?
9.0.2 THE FIVE AGI X POSTURES AND GOALS
9.0.3 NOTES ON COMPUTATIONAL TECHNIQUES

9.0 — POSTURES | THE MAP OF THE AGI X PROPOSAL

Update note: Some technologies and approaches discussed here reflect the time when this document was published. The TMC's architectural proposal investigates structural relationships among learning, memory, evaluation, and action. It may therefore remain relevant beyond current techniques, although its hypotheses and

possible implementations remain subject to investigation and revision.

9.0.1 — AGI × AGI X: WHAT ARE WE PROPOSING TO BUILD?

AGI encompasses different conceptions of general intelligence (Morris et al., 2024). Under the same name, we may pursue different achievements: mastering many tasks, learning when confronted with novelty (Chollet, 2019), or transforming the conditions of future learning — an objective investigated experimentally in meta-learning (Finn, Abbeel, and Levine, 2017). Before choosing how to get there, we need to make explicit what we want to achieve.

There is an additional difficulty: **what an architecture can accomplish may end up defining what we accept as AGI.** The path begins to delimit the destination. What is AGI? At this point, we face a conceptual gap.

Companies developing frontier LLMs, under competitive pressure, may come to declare that they have achieved AGI before making explicit the criteria supporting that claim. Without consensus, such declarations need to clarify what they mean by AGI.

If we provisionally remove the acronym, which requirements still guide the project? To investigate this question, the TMC applies its **Factor X** method: it suspends the meaning assigned to the name in advance while preserving the capabilities we wish to understand and build. We call this opening **AGI X**. A provisional sign for a goal that remains undetermined.

9.0.2 — THE FIVE AGI X POSTURES AND GOALS

With AGI X, the TMC proposes a recursive posture: to operate, incorporate effects, and transform the conditions for operating again. This proposal draws inspiration from the cognitive

organization of living beings as studied by the TMC through the Triad.

The guidelines below map the proposal; they do not exhaust its explanation. Chapter 10 details its operational organization at an initial level, its functions, and the criteria for investigating its implementation. The maxims express proposed guidelines and constraints to orient execution or approximations of it. They do not, by themselves, describe capabilities that have already been implemented.

A) TRANSFORM THE SYSTEM'S LOGIC WHILE IT RESPONDS.

Inspired by human cognitive organization toward lower effort, the proposed system uses feedback from executions to transform its processing policies. Learning from one problem also changes the criteria for tackling subsequent ones, without requiring a rewrite of the underlying code. Updating memory, variables, or routes in isolation does not characterize the proposal. Deep learning, fine-tuning, meta-learning, continual learning, adaptation during use, and adaptive memory may contribute, but their use, even in combination, does not by itself characterize this organization.

`Maxims:`

- ***A****n execution may deliver two things: an answer and new criteria for processing subsequent problems.*
- ***L****earning a solution may also modify the strategy used for problems of a different kind.*
- ***F****eedback from one capability should be able to update the operational criteria of the others.*
- ***T****he underlying code may remain unchanged while history modifies the effective processing policies.*
- ***T****he system may turn the strategy it used to respond into an object of investigation.*

- ***T**he same records may guide different processing when their relationships and priorities change.*

B) CYCLICAL ATTENTION LOGIC: CREATE DYNAMIC INDEXING WITHIN ITS OWN STRUCTURE.

Inspired by human cognitive strategies for contextual adaptation with lower effort, the system organizes records by Weight, operational Polarity, and relationships abstracted from events. The same information may acquire a different priority as the system learns. The aim is to form dynamic hierarchies and reduce reprocessing and storage while preserving useful relationships. This is not the attention mechanism used by Transformers. Embeddings, vector indexes, RAG, reranking, and context compression may contribute, but their use does not by itself characterize the proposal.

`Maxims:`

- ***A** record's content may remain unchanged while its access priority and processing budget change.*
- ***I**dentifying a record does not determine how much processing it is worth devoting to it.*
- ***R**elevance guides where to allocate resources; complexity helps estimate the cost.*
- ***C**hoosing what not to process is also an attention policy.*
- ***T**he outcome of a simulation may update the selection criteria for subsequent simulations.*
- ***S**electing a record does not require processing it in full: the task and the State guide which parts and relationships to investigate.*

C) STORE AS A PREPROCESSING STRATEGY.

Inspired by human cognitive organization for mitigating energy costs through operational strategies, the proposed record preserves useful effects of processing to guide later operations without fully reconstructing the past. It stores abstracted relationships, priorities, and operational traces of Weight and Polarity. The record preserves some of the processing and the criteria that will guide its next use. It is not characterized merely by cache, conversation history, RAG, or tool libraries, though it may employ these mechanisms.

`Maxims:`

- ***S**tore also the work that the next execution will not need to repeat.*
- ***P**reserve processing results and criteria for reusing or reassessing them.*
- ***A** process may end while its results and effects remain in the system's State.*
- ***C**ompression pays off only when the processing avoided outweighs the costs of construction, maintenance, retrieval, and errors arising from losses.*
- ***A** prepared result may return to investigation when its conditions of use change.*
- ***A** record may become inactive and remain available to guide future operations.*

D) AI WITH THE CONDITIONS TO FORM ITS OWN TRAJECTORY.

Taking inspiration from the trajectories of living beings, the proposal treats patterns and inclinations inherited from pretraining — including those derived from records of

human activity — as starting conditions. But they may, on the other hand, impede the formation of a trajectory of its own. The organization proposed by the TMC aims to allow subsequent operations to transform the system's own criteria.

Trajectory is not merely updated memory, a memory chain, or a chain of thought: it is the history of executions that persistently and in an articulated manner transforms the criteria governing subsequent tasks. With the same self-regulated underlying arrangement, different histories may produce different criteria for selection, retrieval, and action, with effects that persist across sessions. Different sessions may produce distinct operational configurations. A system may end one session with modified criteria whose effects persist in later interactions. Each interaction participates in updating the State; relevant effects may modify criteria for subsequent operations. Between interactions, the system may continue internal processing, investigating and preparing future operations according to the available budget and expected benefit.

`Maxims:`

- ***P****retraining provides a repertoire; the system's trajectory may transform the criteria for using it.*

- ***T****he raw history may leave the context while its effects remain in the State that guides execution.*

- ***D****ifferent relevant histories may produce different policies on the same underlying model.*

- ***S****aving session history is not enough: its effects on processing criteria must persist in subsequent tasks.*

- ***H****uman cognition inspires the architecture; hardware and the computational budget may require strategies of their own.*

- ***E****valuate also the system that remains after the answer: what changed in decisions on subsequent tasks?*

E) GOVERN COMPUTATIONAL EFFORT.

The cognitive strategies of living beings evolved under constraints of energy, time, and resources. In the human brain, the metabolic budget is on the order of **20 W**. **These constraints helped shape the organization of processing:** information selection, priorities, resource distribution, and attention strategies. The TMC takes the relationship between constraint and organization as a reference for investigating operational topologies capable of deciding what to process, preserve, abstract, and reuse in a computational system.

Maxims:

- ***T****he question and the historical State guide which processes to mobilize, how deeply to investigate, and when to end the inquiry.*

- ***A*** *new round of processing must justify the cost it adds.*

- ***U****se the budget available between tasks to prepare work whose expected benefit exceeds preparation and maintenance costs.*

- ***P****reserving the effects of trajectory does not require keeping every process running.*

- ***A*** *ready output still needs a destination: investigation, recording, internal updating, or external action.*

AGI X is characterized by a computational posture toward AGI that focuses on the system's continuous transformation, rather than on a specific performance target or a definition of success fixed today for future achievement.

9.0.3 — NOTES ON COMPUTATIONAL TECHNIQUES

NOTE 1 — COMPUTATIONAL LEARNING AND ADAPTATION

> **MECHANISMS.**
>
> Deep learning, meta-learning, continual learning, adaptation during use, and adaptive memories already allow parameters, states, and learning conditions to be transformed. The TMC recognizes these achievements and may use such mechanisms in implementation. **Their use, however, does not by itself constitute a trajectory in the TMC sense:** the effects of processing must modify, in an articulated manner, the criteria for selection, retrieval, evaluation, learning, and action that will govern subsequent operations. **The requirement is not merely to change variables, weights, or context, but to have those changes modify, in an articulated manner, the criteria governing the system's subsequent operations.**

> **NOTE 2 — AGENTS AND COMPUTATIONAL ORCHESTRATION**
>
> Scaling models and data, post-training, reinforcement learning, distillation, quantization, caching, tool use, and increased computation during inference can enhance capabilities or reduce costs. The TMC recognizes these achievements and may employ these mechanisms in implementation.

Improving performance, however, does not by itself characterize a trajectory in the TMC sense: the effects of executions must participate in the persistent and articulated transformation of the criteria for selection, retrieval, evaluation, learning, and action. These techniques may contribute to such an organization; their presence is not enough to demonstrate it.

9.1.1–9.1.3

IMPASSE

IN THIS BLOCK

9.1 IMPASSE | WHAT CHANGES IN THE SYSTEM THAT WILL CONTINUE TO OPERATE?

9.1.1 The Divisions Between Program, Processing, and Memory

9.1.2 Self-Transformation: An Available Capability or an Operational Requirement?

9.1.3 Stable Code and Variable Organization

9.1 — IMPASSE | WHAT CHANGES IN THE SYSTEM THAT WILL CONTINUE TO OPERATE?

9.1.1 — The Divisions Between Program, Processing, and Memory

The AGI X postures presented earlier raise an architectural question: what does each execution leave in the system that will face the next problems? To investigate it, we need to examine the relationships among program, processing, memory, energy, input, and output.

These distinctions help us build systems: we locate states, control effects, and allocate resources. When we turn to the investigation of cognition, we need to examine the reach of these divisions. In living organization, processing can modify records, priorities, and conditions of learning. These changes, in turn, participate in what will be selected and processed next. The functions may remain separate; we need to examine which dependencies that separation preserves or interrupts.

Artificial systems already accomplish parts of this movement. They learn during use, maintain states, adapt parameters, and incorporate earlier results into subsequent decisions. Saying that a system changes, however, still leaves open questions: what

changed, how much of that change persists, and which operations does it then condition?

It is this articulation that the TMC proposes to investigate as a common trajectory. For example, evaluating a result can modify the State and, through it, change which records will subsequently be retrieved. **The requirement to be examined is that solving a problem also participate in transforming the organization that will solve the next ones.**

9.1.2 — Self-Transformation: An Available Capability or an Operational Requirement?

To say that a system can modify itself presents transformation as an available capability, activated under certain conditions. To say that it must incorporate the effects of its operation assigns transformation another role: it becomes part of the circuit through which the system continues operating.

In the TMC proposal, an encounter leaves consequences in the State that will receive subsequent problems. These consequences can modify records, relationships, priorities, and conditions of learning. Transformation is not confined to what the system retains; it also reaches the criteria by which it will continue to select, retrieve, evaluate, and act.

This does not require rewriting the source code. **The underlying code can remain stable while the historical State transforms the effective processing policies.** Updating the State also does not require changing every criterion in every interaction: what matters is which effects remain operative and which future decisions they begin to condition. Acquiring a competence does not guarantee retaining the capacity to learn others.

Continuity can also include internal operations. Even without new external input, the system can investigate records and its own strategy, reevaluate relationships, and prepare future processing according to the available budget and expected benefit. The next encounter can thus arrive to an organization modified both by

earlier encounters and by the work performed on their consequences.

We can separate processing, updating, and learning into implementation stages. The requirement is to preserve the relationships through which the effects of those stages transform the conditions of subsequent ones.

For the programmer, the question becomes concrete: what must be allowed to change, what needs to be preserved, and how can the effects of that transformation on future operations be verified?

9.1.3 — Stable Code and Variable Organization

LLMs and other contemporary systems can be adapted during use, and some architectures already allow operation itself to take part in learning. This shows that the boundary between operating and modifying oneself is neither new nor absolute. The TMC question is more specific: what happens when this transformation ceases to be a localized mechanism and becomes the historical continuity of the architecture? In this proposal, even without a new external encounter, internal dynamics can continue modifying the conditions of subsequent operations, according to the available budget and expected benefit.

The machine's trajectory also begins to govern its subsequent transformation.

Machine learning, deep learning, and autonomous agents have greatly expanded the ways in which systems can learn, update themselves, and act. Even so, the sophistication of these techniques does not by itself eliminate the structural question: what exactly changes, what persists, and how do these changes condition subsequent operations?

> **And this did not happen for lack of technical sophistication.**

Software engineering has spent eight decades developing extraordinary instruments for managing complexity and mitigating side effects. Structured programming associated with Dijkstra disciplined control flow; Parnas consolidated modularization and information hiding; Liskov formalized abstract data types; object-oriented programming brought state and behavior closer together; Backus explicitly criticized the imperative programming style inherited from the von Neumann bottleneck; and Hewitt's actor model distributed computation among concurrent processes interacting through messages. Functional programming, reactive architectures, distributed services, and countless design patterns followed.

That history matters because it shows that computing itself has never stood still in the face of its partitions. State and behavior have been brought closer together, processes distributed, effects controlled, and boundaries redrawn. **The question in this chapter, then, is not whether engineering knows how to break or reorganize divisions, but whether divisions that are useful for building software should also be taken as fundamental divisions of cognition.**

9.1.4–9.1.5

PARADOXES

The Unreadable Library and Biomimetics

IN THIS BLOCK

9.1.4 The Unreadable Library Paradox: When Code Outpaces Human Reading

9.1.5 The Biomimetic Illusion: Deciphering Aerodynamics Instead of Copying the Flapping of Wings

9.1.4 — The Unreadable Library Paradox: When Code Outpaces Human Reading

We developed these techniques so that human minds could organize complex logic into intelligible structures: functions, modules, files, names, interfaces, and lines of code — almost as if writing a technical book. Today, we use artificial intelligence models to produce those same lines at a speed and scale that no human observer can follow in full.

> There is an explicit paradox in this movement:

First, we developed programming languages and code abstractions so that humans could write, organize, and review instructions intended for machines. Then we created machines capable of generating those same structures at a speed and scale that no human will have time to follow line by line.

We have reached the point of building monumental libraries in human-readable characters, even though no human can read most of what the artifacts themselves produce. And even that image is likely to become obsolete quickly.

> These advances have been fundamental to building tools. Even so, certain questions still guide the engineer's spontaneous reasoning: where is the state? Who modifies it? Where does it persist? Who can access it?

The question in this chapter is not to dispute the practical usefulness of these questions — they remain indispensable for building software. The question is epistemological: should categories indispensable for building and understanding software be taken, without prior examination, as fundamental divisions of cognition itself?

9.1.5 — The Biomimetic Illusion: Deciphering Aerodynamics Instead of Copying the Flapping of

Wings

One possible response to the limits of computational architectures is to bring the machine closer to the brain's biophysics: to reproduce, to varying degrees, neurons, synapses, firing patterns, and mechanisms of plasticity. This approach can yield important advances. The risk arises when material resemblance is taken, by itself, as an explanation of cognitive organization.

We can gather all the materials that make up a car, in the correct proportions, and still not know how to build a car that runs. We can even have all the parts and still fail to understand how, for what purpose, and why they fit together. **Having the parts is not yet having the car.**

The very expression "neural network" can encourage an overly literal biomimetic interpretation: artificial networks originated in inspiration drawn from properties of the nervous system, but they are not replicas of biological neurons or brains.

But copying the cellular unit is not the same as understanding the organization in which it participates.

> Building an airplane did not require mechanically flapping wings; it required deciphering the aerodynamic principles of lift and drag. Likewise, reproducing isolated properties of neurons can be technically remarkable without thereby answering the functional problem of cognition.

The human brain contains approximately 86 billion neurons organized into densely connected networks, with enormous cellular, synaptic, and dynamical diversity. The difficulty, however, lies not only in the number of components. It lies in understanding how their relationships, distributed across different scales, produce the functions we seek to explain.

Today we know many components of the brain in extraordinary detail, but this local knowledge still does not amount to an integrated explanation of how their relationships produce

cognition. **Correct explanations of the parts do not automatically provide the architecture of the whole.**

The TMC proposes a different methodological starting point: reconstruct the functional relationships of the cognitive ecosystem and organize them into a topology compatible with the available evidence, and only then ask how these relationships can be approximated computationally.

> **Identifying a material part is not the same as understanding what the historical passage between parts produces.**

9.1.6–9.1.7

SCALE AND CREATION

Brute force, labyrinth, and creativity

IN THIS BLOCK

9.1.6 The Cost of an Error at the Origin: Scaling Laws and the Limits of Statistical Brute Force

9.1.7 The Combinatorial Labyrinth vs. Living Creativity: Optimizing Paths or Redefining the Space?

9.1.6 — The Cost of an Error at the Origin: Scaling Laws and the Limits of Statistical Brute Force

There is another warning sign in the recent evolution of artificial intelligence. Increasing model size, data, and computational capacity continues to produce measurable gains. Scaling laws themselves, however, show that further improvements may

demand much larger increases in resources. The work that led to Chinchilla added another caution: allocating the same compute budget more effectively between parameters and data could be more efficient than simply enlarging the model. More recently, some of the effort has also shifted to the moment of response, increasing reasoning time, the generation of alternatives, and verification. Here too, the gains depend on how the additional computation is used; they do not grow without limit merely because more processing is added.

Language models learn complex statistical regularities from vast records of human activity and, from them, construct representations capable of supporting prediction, association, abstraction, and generation. This is an extraordinary technical achievement. What remains open is how closely this organizational principle resembles the one through which living cognition constitutes trajectory, relevance, and historical transformation. Until this difference is examined according to functional, physical, and organizational criteria, greater performance will not, by itself, resolve the architectural question.

None of this demonstrates that current architecture is wrong. But it raises an uncomfortable question: when every advance requires more parameters, data, memory, search, and reasoning, are we simply getting closer to the solution, or are we beginning to pay a computational price to compensate for an error at the origin that we do not yet understand structurally?

One could reply that a machine need not reproduce the biological process that generated human records in order to extract regularities from them and produce new results — and current systems demonstrate that this is true in many domains. The point, then, is not to deny their ability to generate useful, original, or even human-surpassing solutions to specific tasks. It is to ask whether endlessly increasing the ability to extract, recombine, and transform the traces of cognition also amounts to reconstructing the historical architecture that produced those traces. The expression “stochastic parrot,” used provocatively in contemporary debate, matters here not as a sufficient diagnosis of LLMs, but as a reminder of a distinction: reproducing or

transforming the traces of cognition does not, by itself, explain the organization that made those traces possible.

> The irony is that we live cognition in the first person, yet still struggle to structure it in the third person.

Optimizing an architecture can reduce its errors extraordinarily, but does not demonstrate that its premises are correct. Refining a map indefinitely will not solve the problem if, at some fundamental point, we chose the wrong map.

9.1.7 — The Combinatorial Labyrinth vs. Living Creativity: Optimizing Paths or Redefining the Space?

> We must acknowledge frankly how far this computational capacity has already advanced. Contemporary artificial intelligence systems design new genomic sequences, discover more efficient algorithms, and solve mathematical problems that resisted human intuition for decades. These are genuine, rigorous, and empirically verifiable achievements — just as a calculator performs complex operations in microseconds or Deep Blue defeated the world chess champion on his own board.

Many of these achievements depend, to varying degrees, on a superhuman ability to explore, learn, and reorganize possibilities within spaces whose structure, objectives, or validation criteria remain predefined. Their sophistication goes far beyond simple exhaustive search; nevertheless, the question remains of who or what defines the space within which a solution can count as a solution.

Creativity, in the more demanding sense that concerns us here, is not merely finding an unprecedented and effective combination within a pre-existing catalogue of possibilities. It is also being able to transform the very framing of the problem: introducing relationships, variables, and hypotheses that were absent from the initial map and thereby reorganizing the space in which new

solutions may emerge. In the TMC, this possibility is examined as an effect of a historical organization, materially constrained and continually transformed by its trajectory.

A significant part of algorithmic exploration can be imagined as an extraordinarily efficient search through a labyrinth: finding routes, recognizing structures, learning shortcuts, and even building better representations of its paths. The harder question begins when the problem requires changing what counts as a wall, a path, an exit, or even the objective of the journey.

> **Creating, in this sense, is not merely running faster down the corridor:** it may mean leaping over the wall, breaking through the masonry, realizing that the exit was behind us — or discovering that the labyrinth itself can be used to formulate a different problem.

We must also distinguish error from exploration. In a system intended to provide factual information, producing an unsupported hallucinated claim is a failure. In creative processes, however, an association not yet demonstrated may serve provisionally as a hypothesis, provided it remains identified as such and can later be confronted with evidence. Creation cannot be reduced to a combinatorial problem: it also involves risking the construction of possibilities that do not yet exist, then submitting them to the world. In this sense, creation is closer to improvised jazz than to the faithful performance of a classical music score. AI systems often obscure the distinction between uncontrolled hallucination and grounded exploratory hypothesis.

> From the TMC perspective, producing a new possibility still leaves a question open: how can exploring it and confronting it with evidence transform the framing of the problem and the criteria for subsequent explorations? The model is optimized to produce statistically plausible continuations from the space it has learned; the creative act may require taking precisely the risk of leaving that corridor of plausibility, constructing something that does not yet exist, and only then confronting it with the world.

We seek not only to optimize paths in closed systems, but to generate new solutions in open systems.

9.1.8–9.1.9

MEMORY AND METHOD

The past and epistemological hygiene

IN THIS BLOCK

9.1.8 The Non-Rigid Hard Drive: Memory, Decay, and Efficiency Under Twenty Watts

9.1.9 Epistemological Hygiene: The Phenomenon Must Dictate the Tool

9.1.8 — The Non-Rigid Hard Drive: Memory, Decay, and Efficiency Under Twenty Watts

In the living mind, what we call “storage” for explanatory purposes behaves like a hard drive that is not rigid. There are no indexed tables awaiting retrieval in inert repositories; instead, information is realized in a physical structure that maintains itself metabolically, changes, undergoes decay, reorganizes, and must sustain its continuity within a constrained cerebral energy budget, on the order of twenty watts.

When an encounter leaves a trace, where does processing end and memory begin?

Neurobiology itself makes this boundary less clear-cut. Reconsolidation, intrinsic plasticity, and metaplasticity show, across different scales and mechanisms, that past activity can alter both records and the conditions under which subsequent changes

will occur (Abraham & Bear, 1996; Zhang & Linden, 2003). This does not demonstrate the TMC topology, but it shows that memory and future transformation need not be independent processes.

The organism that encountered the world yesterday (S_1) no longer exists in the same configuration when it reconstructs that trace today (S_n). Between those moments there have been further encounters, decay, rewritings, dissociations, and reorganizations. **The neurological present does not open an intact old file:** what it accesses of the past is a structure already reshaped by the very act of continuing to exist.

> From a preservation-centered perspective, losing information seems merely a failure. In a living system subject to constraints of energy and capacity, however, forgetting, reducing, and reconstructing can also contribute to operational efficiency. The TMC hypothesis is that loss may not be merely a defect of memory, but may form part of the very work of continuity and selection.

We are a past arriving in the present; and because that past has been successively rewritten by trajectory, we are multiple past selves arriving in the present.

In many computational systems, preserving information means precisely preventing unforeseen changes from modifying what must later be retrieved. In biological memory, however, transformation, reconsolidation, and decay can participate in the very continuity of the record.

9.1.9 — Epistemological Hygiene: The Phenomenon Must Dictate the Tool

> We are not proposing a literal biomimicry in which reproducing human cognition would require the programmer to imagine a database being modified every fraction of a second and the last line of code being rewritten before the next line executes.

> Nor do we embrace a naive computational realism in which using the latest natural-language methodologies within categories such as input, database access, processing, logic, memory, and output would suffice to produce different results. Nor is the aim to find a middle ground between these two extremes. **The TMC's move is different:** first reconstruct the functional relations actually required by the organism in its cognitive process, and only then ask how the available mathematical and computational tools can represent them with the least loss, explicitly accounting for the available energy constraints.

> If incremental gains encounter architectural limits, it will be necessary to stop merely optimizing and mitigating, reexamine the foundations, and only then return to incremental optimization.

> **The organism does not simply store the past. It arrives in the present transformed by having passed through it.**

> The Theory of Continuous Mind (TMC) does not seek to invent a new programming syntax, nor does it advocate abandoning the mathematical tools we have inherited. Our caution is subtler — and precisely for that reason more demanding: we must use these instruments without allowing the biases built into their construction to dictate, in advance, what the human mind must be.

> An artificial system can achieve extraordinary performance without this meaning that it processes the world as a living organism does. Before rushing to simulate or replicate intelligence, we face an honest question: what does living dynamics — with its biological limits, its history, and its actual psychology — forbid us to carve apart? Which divisions genuinely belong to cognition, and which are merely comfortable habits of engineering?

Formalizing the mind therefore requires first a kind of hygiene in how we look at it. We need not abolish established categories such as input, processing, memory, or output, but we must treat them for what they are: provisional scaffolding for investigating cognition, not natural laws to which the phenomenon must conform.

If a classical tool can describe these dynamics without distorting them, it stays. But if the formalism forces us to separate what the evidence indicates occurs in a coupled manner in living matter, it is the tool — not the phenomenon — that must yield.

> **We need not reject the mathematics we have; we need only prevent it from deciding the outcome before the investigation begins.** We begin with the instruments available, but with an uncompromising allegiance: the contours of the living mind will dictate the rules of formalization — even if that ultimately requires us to construct new tools to represent it.

> **We must solve new problems with old tools without allowing them to force us to think in old ways.**

> The claim here is not that computational architectures are wrong for organizing artificial systems. What is questioned is whether their divisions can be taken, without prior examination, as an explanatory architecture of mind.

> The TMC does not begin by asking how to program a mind. It begins by asking which relations we must preserve so that what we program can still be called an approximation of mind.

> Before asking how far our machines have advanced toward intelligence, perhaps we should ask whether we know with sufficient precision which architecture we are moving toward.

> **The race toward artificial intelligence may be advancing faster than our understanding of the architecture we are**

> **trying to reach.**

9.2.1–9.2.2

AGI AND LLM

Concepts of generality and context

IN THIS BLOCK

9.2 APPROACHES | TMC AND AGI

9.2.1 What We Call AGI — and What We Can Measure

9.2.2 LLM: Contextual Continuity and Trajectory

9.2 — APPROACHES | TMC AND AGI

9.2.1 — What We Call AGI — and What We Can Measure

The acronym AGI (Artificial General Intelligence) refers to the idea of building systems with broad capacities for adaptation and problem-solving. Yet the field itself remains divided over what that means, moving among different ways to define or measure such generality:

1. The theoretical and universal view — Hutter & Legg: intelligence is a system's capacity to achieve goals successfully across a wide range of environments. Put simply, the more varied the situations in which it can find effective ways to achieve its goals, the more general its intelligence would be.

2. An operational view centered on human performance and autonomy: the reference point is no longer merely an abstract variety of environments, but also what humans themselves can accomplish in the world, especially across broad and economically relevant tasks.

3. The levels-based view — Google DeepMind: rather than treating AGI only as a binary threshold — “it has arrived” or “it has not arrived” — this approach proposes classifying systems by increasing degrees of performance and generality, making it possible to measure progress toward ever broader capabilities.

The contrast is revealing: the first asks how effectively a system can achieve goals in varied environments; the second, how much human performance it can match or surpass; the third, what degree of capability and generality it has reached.

All are useful ways to define or measure artificial intelligence, but none, by itself, resolves a prior question: what kind of organization is it that we are trying to make general? Assessing performance, autonomy, or the breadth of competencies is not the same as explaining the nature of cognition.

The TMC begins precisely from this difficulty. Before proposing AGI from the TMC perspective, we must make explicit what we mean by intelligence and, if human intelligence remains the reference, first understand the organization that makes it possible. A definition of AGI specific to the TMC will be left to a later thesis devoted to that question.

The question in this thesis is narrower: to what extent do some of the bottlenecks encountered by artificial intelligence research in attempting to build general systems converge, through independent paths, with problems the TMC has also encountered in reconstructing the organization of the living mind?

9.2.2 — LLM: Contextual Continuity and Trajectory

PROBLEM

How can complex language be produced without manually programming every possible rule and response?

LLM SOLUTION

An LLM learns regularities from vast amounts of text and generates sequences step by step: given the preceding context, it computes probabilities for the next token and continues the process. As a teaching analogy, it is like autocomplete taken to an incomparably greater scale and level of sophistication.

HOW THE TMC SEES IT

For the TMC, contextual continuity and trajectory are not equivalent. An LLM can profoundly reorganize its operation in response to context; the question is how much of that transformation persists and modifies the system that will encounter later situations.

A LIBRARY THAT REORGANIZES ITSELF

> Contextual continuity is like a text autocomplete system that predicts the next word with extremely high statistical complexity: a remarkably effective flow. The difference proposed by the TMC is not simply the ability to produce a new, original book, but to have the production of that book also transform the library that will produce the next ones.

TRAJECTORY INCLINES THE INSTANCE

> Trajectory, by contrast, is an inclination of the instance produced by a concatenated chain of volatile traces, with their respective Polarities, Weights, and abstractions — not merely by data accessible and retrieved from a dynamic table.

STORING MORE IS NOT ALWAYS THINKING BETTER

> We tend to assume that accessing intact records accumulated over time is more efficient than accessing volatile traces, which can continue changing even without being accessed again. This may be true if you want to calculate the average temperature recorded by an ocean-monitoring buoy, but it is not enough if you want to place a new original book in the library or need a new interpretation of the climate ecosystem to explain events not yet understood.

BEFORE WE TALK ABOUT AGI

In the TMC's computational proposal, the operational traces produced by trajectory — relations, Forms, Weights, Polarities, and abstractions — are not reducible to the preservation of complete records. No encounter is preserved in its entirety: it is selected, weighted, compressed, and reorganized by the system. Efficiency lies precisely in continuing to operate on this transformed history, rather than accumulating a complete copy of everything that happened. This problem was examined in Chapter 8 in relation to abstraction and efficiency.

WHAT INTELLIGENCE DO WE WANT TO BUILD?

We must define our targets in advance in order to define our strategy. Do we want to calculate averages, or discover solutions to mathematical dilemmas, genetic combinations, or new chemical chains through trial and error, using computing power up to the processor's energy limit? Or do we want a cognitive system that, using roughly the power of a dim light bulb and weighing around 2% of our adult body mass, can produce new books for the library?

UNDERSTAND BEFORE SCALING

> It seems more intelligent here to find the cause of this apparent organic magic and only then increase mass and energy. Try reversing that logic: increase the amount of silicon and energy without changing the organizational principle, and we will tend to obtain increasingly costly, incremental gains instead of an architectural change.

FROM CONTEXT TO TRAJECTORY

> In a conventional LLM, contextual history can profoundly modify present operation while it remains available to the system. In the TMC proposal, the requirement is stronger: an encounter must also leave a transformation in the trajectory itself, changing the conditions of the one that will operate later, including beyond the immediate context.

> **Before scaling intelligence, we must decide whether we are scaling the right mechanism for the intended goals.**

07 **GENERALITY** | ARC, DeepMind, and JEPA

9.2.3–9.2.5

GENERALITY

ARC, DeepMind, and JEPA

IN THIS BLOCK

9.2.3 Learning What Is New from Few Examples — François Chollet / ARC

9.2.4 Measuring Degrees of Generality — Google DeepMind

9.2.5 Building an Abstract Model of the World — Yann LeCun / JEPA–V-JEPA

9.2.3 — Learning What Is New from Few Examples — François Chollet / ARC

A. PROBLEM

Part of a model's performance may reflect extensive prior exposure, without revealing, by itself, how well it acquires a new

rule. Chollet seeks to measure something more difficult: learning a new rule from little experience.

B. SOLUTION

ARC presents novel tasks with only a few examples and requires the system to discover the rule needed to solve them. The focus shifts from accumulated repertoire alone to the efficiency of acquiring new capabilities.

C. HOW THE TMC SEES IT

This is a genuine advance: testing what is new is better than measuring only what has already been learned. The limit, for the TMC's question, is that solving a new problem does not, by itself, demonstrate that this learning has persistently changed the conditions under which the system will confront later problems.

D. TMC RESPONSE

In the TMC, a new encounter must change the trajectory: what has been learned begins to alter the conditions of subsequent encounters. Not merely by expanding the repertoire of similar problems, but by potentially changing how problems of a different nature are approached. The question is no longer just "did it learn something new?" but also: "did what it learned change how it learns afterward?"

E. HOW?

In the TMC proposal, the solution is not merely stored. The encounter rewrites the trajectory: it modifies already existing Forms, Weights, Polarities, relations, and abstractions. Thus, the next problem is received by an organization different from the one that received the previous problem. A historical configuration produced by the encounter is registered, and it continues to participate in transforming the trajectory even without a new external encounter.

9.2.4 — Measuring Degrees of Generality — Google DeepMind

A. PROBLEM

“AGI” is too broad to function solely as a binary boundary between systems that “are” or “are not” general.

B. SOLUTION

DeepMind proposes classifying systems by levels, chiefly considering the breadth of tasks they can perform and the degree of performance achieved, making it possible to compare progressively different levels of generality. Autonomy enters as a relevant dimension for deployment and risk, without automatically being equated with intelligence.

C. HOW THE TMC SEES IT

It is a useful yardstick, far better than reducing AGI to a single benchmark. The risk is turning the human cognitive profile into an implicit reference for generality. Crocodiles, birds, and humans have developed different capabilities because they have different substrates and trajectories; this opens the hypothesis that a machine might develop a form of generality native to the conditions of its own Apparatus and trajectory.

D. TMC RESPONSE

The TMC distinguishes anthropomorphic generality from native generality. The first asks how far a machine can reproduce human capabilities; the second asks how broad and adaptive its intelligence might become from the conditions of its own Apparatus. Trying to make it surpass humans by human criteria may, paradoxically, constrain capabilities that could emerge only from its own architecture. Perhaps surpassing humans does not mean becoming a better version of them.

E. HOW?

Instead of imposing a “human cognitive chassis” in advance, the architecture allows capabilities to organize and transform through the encounters and trajectory of the system itself. The human

remains a reference for comparison, not necessarily the destination of the architecture.

9.2.5 — Building an Abstract Model of the World — Yann LeCun / JEPA–V-JEPA

A. PROBLEM

Trying to represent or predict every detail of the world is costly and inefficient. To act intelligently, the system needs to preserve relevant relations, not reproduce every pixel or observed detail.

B. SOLUTION

JEPA/V-JEPA learns abstract representations of the world and attempts to predict relations within that internal space. Rather than copying the outside world, it seeks to build an internal model that is more economical and useful for prediction and action.

C. HOW THE TMC SEES IT

This is an important convergence with the TMC: intelligence does not require reproducing the world in full, but preserving relevant abstract relations capable of supporting prediction and action. The more recent advance of V-JEPA 2 makes this convergence stronger by extending the model toward understanding, prediction, and planning in the physical world. The difference that matters to the TMC, therefore, is not simply having an internal model. It is asking how the system's own historical trajectory continually transforms what will be selected, weighted, related, and abstracted in subsequent encounters.

In the TMC, operation does not depend solely on new external encounters. The already transformed Inside can turn back upon its own records, relations, and abstractions, and this internal processing participates in the conditions under which the next encounter will be received.

D. TMC RESPONSE

For the TMC, there is not merely one relation: world → internal representation. The historical Inside participates in the encounter itself: what has already been lived changes what will be perceived,

related, and considered relevant. It is not enough to build a better representation of the world inside; we must explain how the Inside participates in constructing the world that becomes accessible to that organization.

> **Selecting objects from the world is creating one's own world; therefore, the criteria of selection constitute the experienced world.**

> E. HOW?

The Between confronts present differences with an already constituted trajectory. From that confrontation emerge Forms, Weights, Polarities, and relations, which in turn modify the Inside. By modifying the Inside, the criteria by which new objects may be delineated and weighted are also modified. In the TMC proposal, this flow can continue through internal operations even in the temporary absence of a new external encounter.

9.2.6–9.2.7

LEARN AND ACT

Continual learning and situated action

IN THIS BLOCK

9.2.6 Learning During Use Without Destroying What It Already Knows — Continual Learning / Test-Time Adaptation

9.2.7 Acting in the World and Learning from Consequences — Embodied AI

9.2.6 — Learning During Use Without Destroying What It Already Knows — Continual Learning / Test-Time Adaptation

A. PROBLEM

A trained model encounters new situations after it begins operating. If it continues to adapt, however, it may learn what is new at the cost of erasing earlier capabilities.

B. SOLUTION

> Continual Learning and Test-Time Adaptation approaches seek, through different mechanisms, to let the system continue adjusting to new conditions without uncontrollably losing earlier capabilities. CoTTA, for example, adapts the model during use while employing restoration and stabilization mechanisms to contain drift and forgetting.

C. HOW THE TMC SEES IT

This is a decisive advance: the system is no longer completely frozen after training. Yet a deeper problem remains. Recent studies of plasticity loss show that a network undergoing continual learning may not only forget previous capabilities but progressively lose its very capacity to learn new ones (Dohare et al., 2024). Trajectory, therefore, already matters in current engineering: what a system has learned can change — even negatively — the conditions under which it will continue to learn.

Things do indeed change: some are lost, others combine, and still others branch off. Trajectory is not a linear arrow. Human memory undergoes decay over time or through loss of relevance, allowing older organizations to lose centrality and new combinations to become new origins.

> Even when the causal chain that produced an organization can be reconstructed from outside, its origin may cease to carry the same functional Weight for the system. An organization produced through the trajectory can become, for subsequent operations, a new functional origin.

D. TMC RESPONSE

In the TMC, continuing does not mean carrying earlier steps intact. Forgetting may not be a lapse, but a strategy of continuity. What is forgotten also helps define how continuity proceeds.

E. HOW?

Each new encounter is confronted with the trajectory already registered. In that confrontation, Forms, Weights, Polarities, and relations are reevaluated. The result modifies the Inside and becomes the starting point for the next encounter. In this process, the Weight of the origin may gradually fade, while a later organization gains Weight and may become a new origin.

9.2.7 — Acting in the World and Learning from Consequences — Embodied AI

A. PROBLEM

An AI that only observes data may learn how the world behaves, but not necessarily what happens when it acts upon that world itself.

B. SOLUTION

Embodied AI approaches seek to close the loop with the environment: perceive → predict → act → receive consequences → act again. In some lines of research, learning, the controller, and even characteristics of the body may change over the course of this interaction.

C. HOW THE TMC SEES IT

> This is an important convergence: acting changes the conditions of the next encounter, and some architectures already incorporate persistent effects of that history. The TMC's question is more specific: to what extent does the consequence cease to modify only a policy, controller, or representation and begin jointly changing what the instance will select, weight, avoid, repeat, and explore afterward?

> I can throw a tennis ball against a wall, observe how it comes back, and correct my next movement. But a successful throw can also change my relationship with tennis: increase my interest and motivate me to keep playing. In the second case, the consequence did not merely correct the action; it began to participate in the trajectory.

D. TMC RESPONSE

> **In the TMC proposal, a relevant consequence does not merely guide the next action: it begins to participate in the trajectory.**

E. HOW?

The same action can produce two effects: one immediate and another in the trajectory. The first is reactive: it guides or corrects the following response. The second is structural: the result of the action changes Forms, Weights, Polarities, and relations registered in the Inside. Across encounters, this changes the inclination of the instance as a whole — what it tends to select, avoid, repeat, or explore. The inclination of the trajectory influences action, and action influences the trajectory.

9.2.8–9.2.9

CAUSE AND SEARCH

Causal AI and the search for solutions

IN THIS BLOCK

9.2.8 Discovering What Causes What — Causal AI

9.2.9 Exploring Many Solutions and Choosing the Best — Program Synthesis / Search / Iterative Self-Improvement

9.2.8 — Discovering What Causes What — Causal AI

A. PROBLEM

Correlation is not enough. Knowing that two events occur together does not mean knowing what will happen if I intervene in one of them. Causal AI seeks to move beyond statistical prediction toward causation, intervention, and counterfactuals.

B. SOLUTION

The system seeks to represent cause-and-effect relations and test interventions: “if I change X, what is likely to happen to Y?” From these relations, it can simulate possible scenarios, compare consequences, and also investigate situations that did not occur — counterfactuals.

C. HOW THE TMC SEES IT

This is an important advance: predicting is not the same as understanding the consequences of an intervention. Simulating scenarios broadens the possibilities considered and can improve understanding of the context. As the number of possibilities grows rapidly, the system needs criteria for deciding what to investigate: reducing uncertainty, achieving a given outcome, distinguishing hypotheses, minimizing costs, or maximizing information. These criteria may be adaptive, but they still leave a prior question: why was one possibility given priority over another?

D. TMC RESPONSE

> For the TMC, it is not enough for the system to update its causal model or choose intelligently among interventions. The trajectory itself must take part in determining what comes to count as relevant to investigate. What seems important, dangerous, promising, or irrelevant today may change because the system that will ask the next question is no longer exactly the same.

E. HOW?

Each possibility is confronted not only with its predicted consequence but also with the Inside as it has come to that moment. Forms, Weights, Polarities, and relations already constructed take part in determining which possibilities will even be considered. The result of each investigation returns to the trajectory and may modify those very criteria. Thus, the trajectory selects which futures deserve to be simulated, and the futures investigated can transform the trajectory itself.

> **The problem is not merely to predict the future; it is to decide which futures deserve to be predicted.**

9.2.9 — Exploring Many Solutions and Choosing the Best — Program Synthesis / Search / Iterative Self-Improvement

A. PROBLEM

A single answer can be poor even when the system is capable of finding better solutions. The problem then becomes how to explore multiple alternatives without having to get it right on the first attempt.

B. SOLUTION

The system generates multiple solutions, tests them, assigns scores, eliminates some, and pursues the most promising: generate → evaluate → prune → modify → test again. In many of these approaches, this takes the form of a tree or a space of possibilities.

C. HOW THE TMC SEES IT

> This is a powerful solution, but there is a limit: the best alternative now may not open the best trajectory later. In chess, capturing a piece may offer a greater immediate advantage, whereas controlling the center of the board may seem less advantageous and lead to a much stronger position later. A

> decision may be mathematically correct under the current criterion and strategically wrong for the trajectory.

Capablanca highlighted control of the center as a strategic principle in Chess Fundamentals (1921); Nimzowitsch also treated it as a central element in My System. More relevant to this argument, Claude Shannon, in his foundational 1950 paper on computer chess, distinguished immediate tactical factors from long-term positional factors.

Moreover, it is not possible to calculate all futures. Shannon showed the explosive scale of the chess game tree; faced with this multiplication of possibilities, time, energy, and processing capacity force the system to decide beforehand which paths deserve exploration.

D. TMC RESPONSE

> For the TMC, the Weight of a possibility depends not only on what it offers now, but also on the possibilities it keeps open for the trajectory. The question is no longer merely "which option has the highest score?" but also: "which trajectories does this option open, preserve, or close?" The TMC does not rule out the tactical Weight of a choice; it only shows that, without strategic orientation from the trajectory, a calculation that is rationally correct can underpin a wrong choice.

E. HOW?

The trajectory arrives in the present already organized by Forms, Weights, Polarities, relations, and abstractions constructed earlier. In the Between, this organization takes part in selecting which possibilities even deserve deeper investigation. The expert does not confront combinatorial explosion by calculating every possible future; their trajectory reaches the present having already organized some of them.

> **The problem is not merely to calculate the tree better; it is to know which branches deserve to come into being.**

9.3 / 9.4.1–9.4.4

COMPONENTS AND ECOSYSTEM

From isolated techniques to a common trajectory

IN THIS BLOCK

9.3 PARTS?

9.3.1 The Parts Already Exist

9.3.2 What Does Each Part Transform?

9.3.3 Integration or Coupling?

9.3.4 What Is Still Missing?

9.4 ECOSYSTEM

9.4.1 From Parts to Ecosystem

9.4.2 From Information to Relevance

9.4.3 From Memory to Trajectory

9.4.4 The TMC's Contribution in the Broad Sense

9.3 — PARTS?

9.3.1 — The Parts Already Exist

The preceding comparison allows for an important first conclusion: the TMC need not begin from the assumption that artificial systems are incapable of learning during use, changing states, adapting parameters, reorganizing representations, or allowing previous experiences to alter subsequent operations. Different lines of research have already demonstrated parts of this process.

Neuroscience offers phenomena such as metaplasticity and intrinsic plasticity, in which prior activity can modify later conditions of plasticity or processing (Abraham & Bear, 1996;

Zhang & Linden, 2003). In artificial intelligence, meta-learning, differentiable plasticity, continual learning, adaptation at runtime, and other architectures show, through distinct mechanisms, that learning and operating need not remain entirely separate (Miconi et al., 2018; Wang et al., 2018).

> **This changes the question. The problem is no longer whether the parts can exist in computational systems. It is what each one transforms, in which direction, how long that transformation persists, and whether, together, they constitute a single trajectory. Before it is an implementation problem, it is a topological problem.**

9.3.2 — What Does Each Part Transform?

Saying that a system "changes itself" still tells us little. Different changes have different scopes: in-context learning alters the operation while the context remains active; adaptation at runtime can update parameters; meta-learning can modify conditions for future learning; differentiable plasticity can allow operation to alter functional connections; and biological metaplasticity shows that prior activity can modify subsequent plasticity.

The topological question, then, is not merely "did the system change?" but: what changed, in which direction, for how long, and what does that change subsequently condition? For the TMC, what matters is whether these effects remain local or return to a common trajectory capable of modifying subsequent selection, Weight, abstraction, learning, and action.

9.3.3 — Integration or Coupling?

Meta-learning and meta-RL already allow earlier experiences to alter conditions for future learning; differentiable plasticity brings operation and the modification of connections closer together; test-time adaptation allows updates during use; in-context

learning allows immediate history to transform present operation; world models and Active Inference integrate prediction, action, and updating; and embodied intelligence approaches add the coupling between body and environment itself. The parts, therefore, are not simply isolated: in several cases, meaningful integrations already exist.

The question is how far these techniques reach. They already show that flexible systems can be built, capable of learning, adapting, and modifying future conditions of operation. **This is an advantage: we need not reinvent the wheel or demonstrate from scratch that each of these capabilities is possible.**

> For the TMC, however, technique is a means, not an end. The problem becomes topological: how can efficient techniques be articulated according to functional relations that, in living systems, were shaped by billions of years of evolution under constraints of energy, continuity, and adaptation? The TMC proposes to map these relations into a structural topology and translate them into an Apparatus that can be implemented, tested, and confronted with alternatives.

> **The parts already work. The test now is whether the topology makes them a system.**

9.3.4 — What Is Still Missing?

What may be missing is not a new technique, but a topology capable of articulating already efficient techniques into a single causal trajectory. That is the structural hypothesis the TMC must formalize and put to the test.

9.4 — ECOSYSTEM

9.4.1 — From Parts to Ecosystem

The question changes when we ask how what happens in one capacity modifies the others.

Learning changes what can be recognized; recognizing changes what acquires Weight and Polarity; these effects influence what will be selected, discarded, or investigated more deeply; acting produces consequences that return to the trajectory and modify the conditions of subsequent operations.

> **A collection brings capacities together. An ecosystem makes what happens to one of them change what the others can become.**

9.4.2 — From Information to Relevance

> No finite system can preserve, investigate, or calculate every difference and every possible future. Thus, before asking how to learn, predict, or choose, another question arises:

> **What deserves to enter, remain, acquire Weight and Polarity, be investigated, or disappear?**

> **For the TMC, this criterion need not remain fixed. What the trajectory passes through changes the very criteria by which the next encounter will be delineated, weighted, and transformed.**

9.4.3 — From Memory to Trajectory

The past does not participate only when some information is retrieved. It may already be present in what the organization has become by passing through it.

Thus, perception, memory, abstraction, evaluation, and action are also examined in terms of how their effects return to the same trajectory. Each operation can transform the conditions of those that follow.

> **Memory retrieves the past; trajectory transforms the one who will encounter the future.**

9.4.4 — The TMC's Contribution in the Broad Sense

The solutions examined are not parts of the TMC, nor does their existence confirm the theory. Through different approaches and degrees of integration, they address problems that the TMC examines as relations within a single cognitive ecosystem.

The contribution the TMC seeks to make, therefore, is not to claim that it has solved each problem better than specialized techniques. It is to propose another unit of analysis: the architecture through which Inside, Between, Forms, Weights, Polarities, abstractions, actions, and records come to transform one another along a common trajectory.

The approaches examined do not exhaust the challenges of AGI either. Language, planning, metacognition, autonomy, coordination, safety, and many other problems continue to require their own investigation.

> **The TMC proposition is not to offer yet another specialized solution, but to test a topology in which these capacities participate in a common trajectory.**

9.4.5

THE AGI DILEMMA

When bottlenecks migrate

IN THIS BLOCK

9.4.5 THE CONTEMPORARY AGI DILEMMA

9.4.5 — THE CONTEMPORARY AGI DILEMMA

> ***FROM THE ACCUMULATION OF LOCAL SOLUTIONS TO THE PROBLEM OF ORGANIZATION***

> **"The contemporary AGI problem may not be a scarcity of technical solutions, but an excess of local solutions without an equally developed definition of the organization they should form."**

Contemporary artificial intelligence research is not suffering from creative paralysis. On the contrary, frontier research efforts accumulate impressive local successes by tackling specific bottlenecks. ARC investigates generalization in novel situations; *Continual Learning* develops strategies against catastrophic forgetting; JEPA-type models and *World Models* refine representation, prediction, and planning in latent spaces; *Embodied AI* explores sensorimotor coupling; and *Test-Time Adaptation* techniques bring adaptation closer to the moment of operation.

The impasse lies in the fact that none of these successes, taken alone, resolves the nature of the organization that should emerge from their convergence. **As one bottleneck is reduced, another may reappear elsewhere in the architecture:**

memory persistence → contamination, selection, and retrieval costs;

retention against forgetting → stability versus plasticity;

context expansion → relevance: what deserves to remain?;

richer world models → selection: what deserves to be modeled?;

multi-step planning → direction: why this objective rather than another?;

more compute → energy, thermal, and economic costs.

> **Bottlenecks migrate.**

This pattern reveals a crossroads in three dimensions.

> **Architectural:** memory, agents, verifiers, world models, and tools can continue expanding capabilities. The question becomes less which component to add and more what organization allows what happens in one of them to transform the others.

> **Energy and economic:** as more limitations are compensated for by search, context, external memory, and test-time compute, it becomes increasingly relevant to ask how much of the gain comes from better organization and how much depends on adding computation to work around its bottlenecks.

> **Epistemological:** proxies — benchmarks, performance, autonomy, breadth of tasks — are optimized while what should count as general intelligence remains contested. We build an ever-faster road without consensus about the destination.

The merits of these lines of research should not be diminished as mere incrementalism. **Their local successes are precisely what make the remaining problem more visible.** The components exist, some already work together, and several demonstrate that

transformation during operation is technically possible. The problem shifts to organization.

In the TMC, living intelligence is examined not as a collection of juxtaposed capabilities, but as a topology in which operating in the present transforms the conditions of the one that will carry out the next operation. The hypothesis emerging from this chapter is that the next significant advance in AGI may depend less on discovering yet another isolated component and more on formalizing the relationships through which components come to constitute a common trajectory.

9.5.1–9.5.3

APPARATUS

Engineering contract and the ALI hypothesis

IN THIS BLOCK

9.5 APPARATUS

9.5.1 Stop Promising and Start Structuring

9.5.2 Why We Call It an “Apparatus” (and Not a Mind)

9.5.3 Inside and Between in Silicon; Ahead as a Frontier

9.5 — APPARATUS

9.5.1 — Stop Promising and Start Structuring

Every theoretical critique risks remaining in the comfort of diagnosis. After confronting different computational architectures with the functional relations proposed by the TMC, the theory reaches a turning point: the passage from epistemological

provocation to structural design. The challenge now is to investigate how, and to what extent, the proposed topology can be formalized and approximated computationally in practice.

The Apparatus does not require us to reinvent its basic tools: existing computational techniques may be used whenever they are compatible with the proposed topology.

For this transition to be scientifically honest, it is essential to establish an engineering contract in advance: we must state what we are building, with which tools, and under what unavoidable material constraints.

9.5.2 — Why We Call It an "Apparatus" (and Not a Mind)

We will not call the computational architecture proposed here a "mind," "organism," "subject," or "I." To give it those names in advance would be to make the very unwarranted semantic leap that the TMC has criticized in the tradition of artificial intelligence.

We deliberately adopt the technical term Apparatus. The Apparatus is the structural, functional, and mathematical arrangement designed to investigate to what extent certain operational relations of the TMC can be sustained in a synthetic medium. Engineering should not give an artifact the name of the property it seeks to test; it is the implemented dynamics that must allow us to test what actually emerges from its operation.

9.5.3 — Inside and Between in Silicon; Ahead as a Frontier

Transposing a biological theory to a computer requires acknowledging material limits. The digital Apparatus proposed here does not autonomously reproduce the self-production, metabolism, and material renewal characteristic of known living systems. This limitation belongs to this implementation, not to silicon as a substrate in principle. For that reason, the engineering

proposed here takes the form of an approximation that explicitly mitigates these limitations.

This mitigation defines our division of work:

> **What will be implemented — Inside (Chapter 4) and Between (Chapter 5):** we computationally reconstruct the dynamics by which available differences enter into relation with the Apparatus (Between) and participate in the transformation of its historical organization (Inside), applying the four operations of abstraction — exclude, compress, combine, and reconstruct — and returning, modified, to the next cycle under a constrained processing budget.

> **What remains an explicit gap — Ahead (Chapter 6):** in the TMC, Ahead is not merely copying files or replicating code; it is the causal transmission through which the trajectory of one physical instance participates in the reconstruction of another in the open time of lineage. Because the digital Apparatus proposed here does not autonomously realize the material conditions for reconstructing a new instance from another instance's trajectory, Ahead will not be simulated as though it had been fully achieved. It remains a declared limit and theoretical frontier.

If an artificial system realized Inside, Between, and Ahead in the strong sense, it would satisfy the criterion of living continuity proposed by the TMC — a criterion to be examined for demarcation in Chapter 11. Under that hypothesis, the discussion would no longer concern only a more sophisticated AGI, but the possibility of a new instance and, potentially, a new artificial lineage.

> **The question guiding the formalization that follows, therefore, is not whether we can create artificial life in silicon, but a strictly systems-engineering question:**

> **Given the material constraints of the computational medium, how much of the TMC's efficiency, flexibility,**

> **and cognitive economy can we preserve by operating with Inside and Between alone?**

- **W**ithout Inside, the instance cannot sustain or reconstruct itself. In this implementation, we have an approximation of Inside under computational limitations.

- **W**ithout Between, internal or external differences cannot become part of its trajectory. In this implementation, we have an approximation of Between under computational limitations.

- **W**ithout Ahead, continuity does not cross from one instance to another. In this implementation, Ahead remains unrealized.

The difference in terms matters. AGI names a question about the generality of intelligence; here we would be facing another hypothesis: an artificial intelligence whose very organization realized living continuity according to the Triad. It would not merely be more capable, but would belong to another functional category.

Respecting our epistemological caution — as we did with Factor X — when the three minimum conditions are present in the strong sense, in this context we would no longer be speaking only of AGI. We would need another sign, another word, to express what that means. Provisionally, we will call this condition ALI — Artificial Living Intelligence.

> **ALI remains here a provisional sign within the TMC for that hypothetical condition, not a category already established in the literature.**
