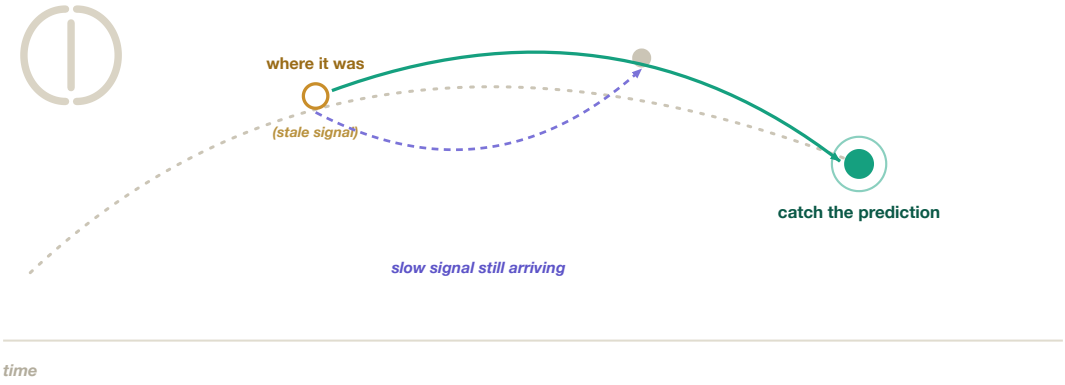
Cere

How a tiny, fast model learns to think ahead of a slow one — and what happened when we ran the idea for 200 experiments.



RESEARCH MONOGRAPH

Cere — An Online-Learning Cognitive Architecture

Online speculative action execution: pairing a slow, deliberate model with a tiny, fast one that predicts the next action before the slow model finishes — and learns, online, to do it well.

Status. All 200 research phases across 39 arcs are built and **GO** on minimal, reproducible substrates *in isolation*; the open work is folding them into one continuously-learning runtime.

Date 2026-06-26 · **Machine** Apple M2, 16 GB · **Code** software-only, NumPy with a small real-model track (Qwen2.5-0.5B distilled from Claude-Haiku-4.5).

Companion documents. VISION (why) · cognitive-systems (faculties) · cognitive-map (arc order) · ROADMAP (plan) · STATUS (live state) · experiments (evidence) · decisions/ (architecture decision records).

> **A note on this document**
>
> This is a self-contained account of the Cere research, written to be read top to bottom by someone who has never seen the project and who has no machine-learning background. Every major idea is introduced three times — once as an *intuition* (an analogy you already understand), once as a *mechanism* (what the computer actually does), and once as *evidence* (the experiment that tested it). Read only the boxed intuitions and the diagrams and you will still understand the whole project; read everything and you will understand why each piece had to exist.

ABSTRACT

Abstract

Large language models are smart but slow. Today's AI agents pay the full cost and the full delay of a big model on every action — including the routine ones a competent assistant would handle without a second thought.

Cere (from *cerebellum*) tests a single idea borrowed from how your own brain hides delay: pair the slow, deliberate model with a **tiny, fast model that predicts the next action before the slow one has finished deciding** — and let the fast model **learn online** from two teachers: the slow model's eventual decision, and what actually happened in the world. Over time, routine actions migrate from slow-and-deliberate to fast-and-reflexive, the way a practiced skill becomes automatic. The big model stops being the executor and becomes a **court of appeals** — consulted only on novelty, uncertainty, risk, or dispute.

The project ran this idea through 200 experiments. It began as a latency trick (predict ahead, act now) and grew, one tested mechanism at a time, into the broad **cognitive architecture** described here: perception, fast cognition, slow cognition, value, planning, memory, learning, simulation, verification, safety, consolidation, and — at the far end — the machinery for the system to notice which cognitive faculty it is *missing* and propose building it, behind a human-approval gate. Every mechanism *presented in this work* was evaluated in isolation on a small, reproducible substrate against named baselines, with an explicit pass/fail criterion; the headline claim was additionally tested with real models, where a 0.5-billion-parameter model learned a frontier model's labelled behaviour from its outputs alone, reaching the teacher's *measured* competence ceiling on a single laptop. The pieces are evaluated individually; assembling them into one continuously-running system is explicitly *not* claimed here, and is named throughout as the central open work.

This report explains what Cere is, why it exists, how it works, what has been proven, what has not yet been built, and where it goes next.

200
phase-experiments, across 39 research arcs

0.09→0.91
tiny model learns a frontier model's behaviour (real models)

0
unsafe irreversible commits, across the adversarial tests run

~90,000×
fast-path forward-pass headroom under the latency budget

§ Reading the headline numbers

So none of these is ambiguous on first sight: **0.09 → 0.91** is top-1 action-prediction accuracy of a real 0.5B-parameter student (Qwen2.5-0.5B) before vs. after distilling a frontier teacher (Claude-Haiku-4.5) on opaque action codes — 0.91 was the teacher's own accuracy ceiling, so the student reached it (Part IV; Faculty 1). **∼90,000×** is the ratio of the latency budget (≈1000 ms) to the fast model's measured forward pass (≈0.011 ms) — headroom under the budget, *not* an end-to-end wall-clock speedup (Part III F1; Appendix B.3). **0** is the count of unsafe irreversible actions that passed the safety gate across every adversarial probe we ran on the evaluated substrates (e.g. 332/332 and 350/350 catches; Part II §5). **200** is defined next.

§ What “200 experiments” means — read this before the number

The number is a unit of *methodology*, not a marketing figure, so it is worth pinning down up front. A **phase** is one self-contained experiment: a single mechanism, built on the smallest substrate that can express it, run against named baselines, with an explicit “done-when” criterion fixed in advance. There are **200 phases** because the project advanced one mechanism at a time and never merged two into a single phase. The phases are grouped into **39 arcs** (plus capstones) — each arc is one research question that drops a comforting assumption the previous arc relied on (a correct teacher, a checkable answer, a short horizon). So “200 experiments” is better read as **39 hypotheses, each tested by a short ladder of phases, with the arc’s capstone phase as its decisive test**. Part IV tells that story in full; the master map (Table 3) lists every phase against its arc. Crucially, a phase being *GO* means it met its own criterion *in isolation* — it does *not* mean the 200 mechanisms have been run together at scale.

CONTRIBUTIONS & SCOPE

Contributions

This work contributes the following, each stated so it can be checked against the evidence later in the report:

1. **Online speculative action execution.** A draft policy that *learns online* (rather than staying frozen as in speculative decoding) and acts at the level of *actions*, hiding a slow model's latency. *Evidence:* a learned predictor reaches 74% task success where wait-and-react reaches 1–3% at the same latency budget (Part III, Faculty 1; Part IV, Arc I).
2. **Speculative cognition.** The identical consolidation machinery, applied to *reasoning*, makes routine reasoning reflexive — the *same transfer-library code*, byte-for-byte, consolidates both actions and reasoning. *Evidence:* −45% decisions (actions) and −62% steps (reasoning) at held quality (Part III, Faculty 3; Part V).
3. **A learned, calibrated, moving routing boundary ("jurisdiction").** A calibrated value head and a learned wrongness head replace a hand-tuned routing threshold. *Evidence:* the value-informed gate catches 62% of committed errors at a 10% escalation budget vs. 46% for policy confidence; the wrongness head reaches AUROC-of-error 0.983 vs. 0.749 (Part III, Faculties 4 and 6).
4. **A sealed reversibility/verification gate that holds under composition and self-modification.** Safety is a separate mechanical step, not a per-faculty promise. *Evidence:* 0 unsafe irreversible commits across adversarial probes (e.g. 332/332, 350/350), with the full regression suite holding simultaneously at every capstone (Part II §5; Part V; Part III, Faculty 9).
5. **Robust distillation that can exceed an imperfect teacher.** Inferring the latent principle (not cloning behaviour) lets the student surpass a noisy teacher and survive its replacement. *Evidence:* 0.814 vs. the teacher's own 0.714, transferring across a teacher swap with zero re-learning (Part III, Faculty 7).
6. **A real-model demonstration that the core mechanism survives contact with real models.** *Evidence:* a 0.5B-parameter student reaches a frontier teacher's accuracy ceiling (0.09 → 0.91) from labels alone, on a laptop, in ~4 epochs (Part IV).
7. **An isolation-first, pre-registered-criterion methodology** spanning 200 phases / 39 arcs, with named baselines and byte-level reproducibility at the capstones. *Evidence:* two independent capstone runs hash identically (Part IV; Appendix B).

Scope of claims — what this work does and does not assert

THIS WORK DEMONSTRATES

- ✓ a tiny learning model can take over a slow model's *routine* actions and reasoning, safely, on the evaluated substrates;
- ✓ the routing boundary between fast and slow can be *learned and calibrated*, not fixed;
- ✓ a *sealed* gate keeps unsafe irreversible commits at zero under adversarial probes and under composition of all faculties;
- ✓ the core mechanism holds on *real models* (0.5B student ← frontier teacher).

THIS WORK DOES NOT CLAIM

- ✗ AGI, or human-level general intelligence;
- ✗ production deployment — all ~200 mechanisms running as one live system at scale with live large models (this is the central open work);
- ✗ that it *replaces* large models — the big model remains teacher, verifier, and court of appeals;
- ✗ fidelity to human or biological cognition — the brain analogies are design hints, not claims of replication;
- ✗ results on external, third-party benchmarks the project did not design (not yet run).

ORIENTATION

How to read this report

	Part	What it covers
Part I	The Idea	Why Cere exists, and the one analogy that explains it.
Part II	The Architecture	The whole system on one page, then each piece — and how it grew.
Part III	The Faculties	Each cognitive “organ”: what problem it solves, how, and the proof.
Part IV	The Experiments	The 200-phase story, told as 39 research questions and a map.
Part V	The Learning Loop	The one loop that never changed across all 200 phases.
Part VI	Research → Runtime	CereOS: turning proven cognition into a living system.
Part VII	Proven vs Not	The honest ledger — what is real, what is still a sketch.
Part VIII	The Future	How the next research integrates, rather than replaces.
Appendix	Reference	Glossary · every phase mapped to its arc · how to reproduce.

◐ **Three reading paths**

Five minutes — read Part I and the two large diagrams in Part II.
One hour — add Part IV (the experimental story) and Part VII (the honest ledger).
Evaluating the research — Parts III, V, and VII are where the claims and their limits live.

Contents

PART I

Why Cere exists, and the one analogy that explains it

The Idea

Today's agents pay genius prices and genius delays for work a junior assistant could do instantly. Cere's whole project is one elaboration of a single picture: you do not catch a ball where it is — you catch where it is going to be.

> *Three-quarters of all your neurons sit in the cerebellum, running a fast physical imagination that lets you act before your conscious mind catches up.*

1 The problem: smart but slow

Imagine hiring a brilliant consultant who is also extremely deliberate. Every question you ask — even *"should I file this receipt under March?"* — gets the same treatment: they lean back, think for ten seconds, and give you a thoughtful answer. For the hard questions, that thoughtfulness is exactly what you want. For the hundred routine ones, it is agony. You are paying genius prices and genius *delays* for work a junior assistant could do instantly.

That is precisely how today's AI agents work. A large language model — an *LLM*, the kind of model behind modern AI assistants — is genuinely smart, but each decision takes meaningful time and money. Wire one up as an *agent* — a system that takes actions, calls tools, clicks buttons, sends emails — and it pays the full cost of that big brain on **every single step**, routine or novel.

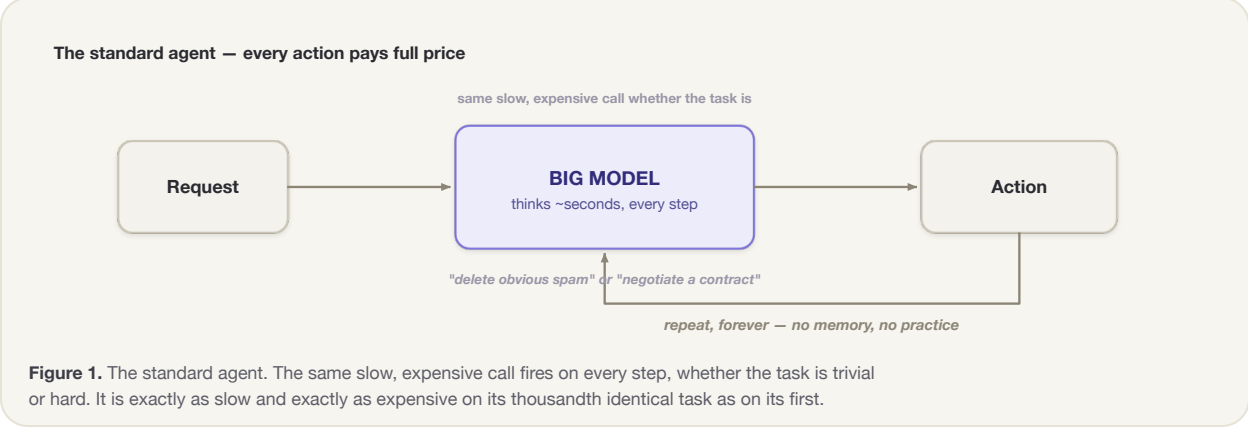


Figure 1. The standard agent. The same slow, expensive call fires on every step, whether the task is trivial or hard. It is exactly as slow and exactly as expensive on its thousandth identical task as on its first.

Two costs compound. **Latency:** the world does not wait for a slow thinker, and a chain of slow steps becomes a painfully slow workflow. **Money and competence misallocation:** the expensive model is spent on the easy 90% instead of being saved for the hard 10% where it earns its keep. And a third, subtler loss: the system **never gets faster at things it has done a thousand times.** Every call starts from zero. There is no such thing as practice.

2 The analogy that explains everything: catching a ball

Here is the move Cere makes, and it is worth slowing down for, because the entire 200-experiment project is one elaboration of this single picture.

◐ You do not catch a ball where it is

When you catch a baseball, you do not wait for the ball to arrive and then move your hand. By the time you *saw* where the ball was, it had already moved — the signal from your eyes to your conscious mind is too slow. Instead, a fast, unconscious part of your brain **predicts where the ball is going to be** and sends your hand there in advance. You catch a *prediction*, not a ball.

The part of your brain doing this is the **cerebellum** — about three-quarters of all your neurons. On the leading account it runs a fast *forward model* (**Wolpert et al., 1998**; **Miall et al., 1993**) — a "physical imagination" that anticipates the next fraction of a second so you can act before your slow, deliberate, conscious mind catches up. And it **learns**: the first time you try a new sport you are clumsy and have to think about every motion; after practice the same motions become reflexive. Deliberate effort, repeated, turns into reflex.

Cere brings exactly this split to software. Pair the slow deliberate model (the "conscious mind") with a tiny fast model (the "cerebellum") that predicts the next action *now*, and let the fast model **learn from practice** so that routine actions become reflexive — handled instantly, locally, for nearly free — while the slow model is reserved for the genuinely novel.

Relation to predictive-processing theories

The "predict ahead to beat delay" idea has a deep lineage in neuroscience — predictive coding (**Rao & Ballard, 1999**), the predictive brain (**Clark, 2013**), and the free-energy principle / active inference (**Friston, 2010**). Cere borrows the *intuition* (a forward model anticipates the near future so action need not wait on a slow signal) but, to be precise, it does **not** implement active inference or free-energy minimization: there is no variational objective and no unified perception-action inference loop. Cere's predictor is a small supervised model trained on a slow teacher's decisions and real outcomes. The relationship is conceptual inspiration, not implementation.

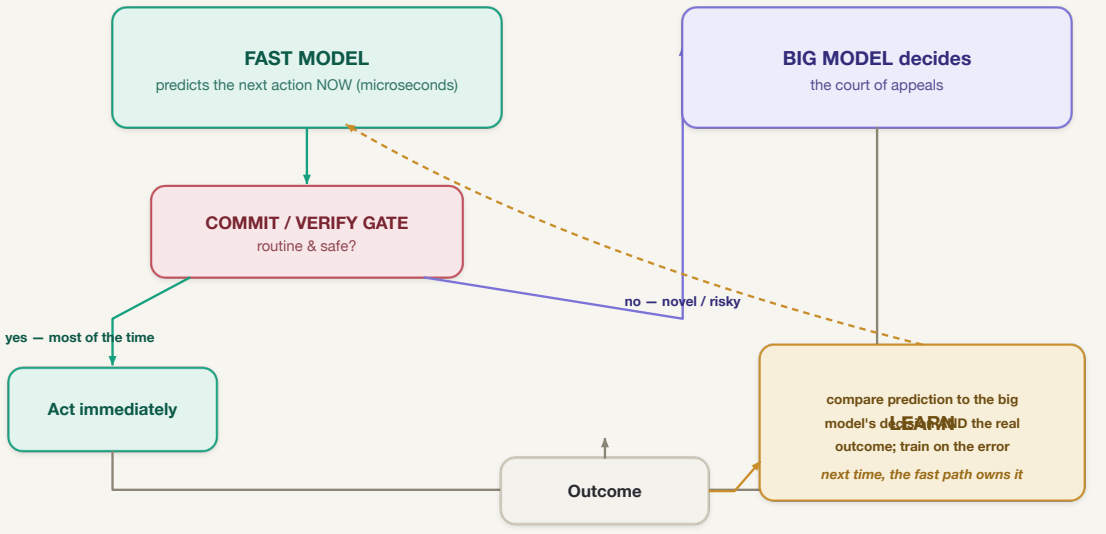


Figure 2. Cere: predict ahead, verify, and learn from practice. The fast model proposes an action in microseconds; a gate decides whether to trust it or appeal to the big model; the outcome feeds a learning step that hands the case to the fast path next time. The name for this, in one phrase, is *online speculative action execution*.

The fast model is not a smaller, dumber chatbot. It is a *different kind of thing*: small enough to run in well under a millisecond, and trained continuously on the slow model's decisions and the world's feedback.

3 The thesis, stated precisely

The research community already has a trick called **speculative decoding** (**Leviathan et al., 2023**; **Chen et al., 2023**): a small model guesses several next words, and the big model checks them all at once, which is faster than generating each word from scratch — a *lossless* inference speedup in which the draft model is **frozen** (see also blockwise parallel decoding, **Stern et al., 2018**). Cere's thesis is a deliberate twist on that idea.

> **§ Thesis (falsifiable)**
>
> Speculative decoding, but **(a)** the draft model **learns** instead of staying frozen, and **(b)** it operates at the level of **actions and reasoning** instead of individual words. A fast "draft" policy that learns online — from the slow model's eventual decision *and* from the real outcome — will let routine actions migrate from slow/deliberate to fast/reflexive, approaching zero-latency behaviour on learned actions, while novel cases fall back to the slow model.

Two value propositions follow, and the project tested them in order:

- **Latency** (the first three experiments): the fast model predicts ahead and acts now, hiding the slow model's response time. *Does prediction actually beat just waiting?*
- **Cost & competence** (the fourth experiment, and everything after): the slow model stops checking every step and becomes a **graduated supervisor**. *If the fast path is 90% competent, the big model only handles the hard 10%, so cost tracks task **difficulty**, not request **volume**.*

The big model is no longer the default executor. It is the appeals court.

That reframing is the spine that runs through all 200 experiments.

What is inherited, and what is new (the short version)

None of the ingredients is novel on its own, and the report does not pretend otherwise. A small draft model verified by a big one is *speculative decoding*; routing easy cases to a cheap model and hard ones to an expensive model is a *model cascade*; spending more compute only when needed is *adaptive computation*; learning without forgetting is *continual learning*; a value network plus search is *AlphaGo*. Cere's claim is narrow and specific: it is the **combination** — a draft policy that *learns online*, at the level of *actions and reasoning* rather than tokens, with the routing boundary itself *learned and moving*, all inside a *sealed reversibility/verification gate*. **Appendix D** compares Cere head-to-head with each of these areas — what each solves, what Cere inherits unchanged, what it changes, and what remains future work.

Where Cere sits in the history of AI

It helps to place the idea on the timeline. Symbolic *expert systems* hand-wrote rules; *reinforcement learning* (**Sutton & Barto, 2018**) learned policies from reward; *deep learning* learned the representations those policies run on; *large language models* built on the transformer (**Vaswani et al., 2017**) turned broad competence into a single pre-trained model; and *LLM tool-agents* (**Yao et al., 2022**; **Schick et al., 2023**) wired that model to actions. Each step added capability but kept one habit: at run time the big model does all the work, and nothing it does makes the next call cheaper. The most recent inference-time idea, *speculative decoding* (**Leviathan et al., 2023**), breaks that habit for *tokens* with a frozen draft. Cere asks what happens if the draft is not frozen but **learns online**, and operates on **actions and reasoning**. It does not replace any of these paradigms — it adds an online-learning fast path *beside* the LLM agent. It is an evolution of the agent loop, not a successor to it.

4 Cere versus a standard LLM agent

Almost everything distinctive about Cere is a box that the standard picture simply does not have. The standard agent observes, thinks, answers — and is exactly as fast and exactly as expensive on its thousandth task as on its first. Cere adds *prediction* (act before the slow model finishes), *verification* (a safety boundary deciding when to trust the fast path), and an entire **learning spine** — learn, remember, consolidate, improve — so that the system gets faster and cheaper with use.

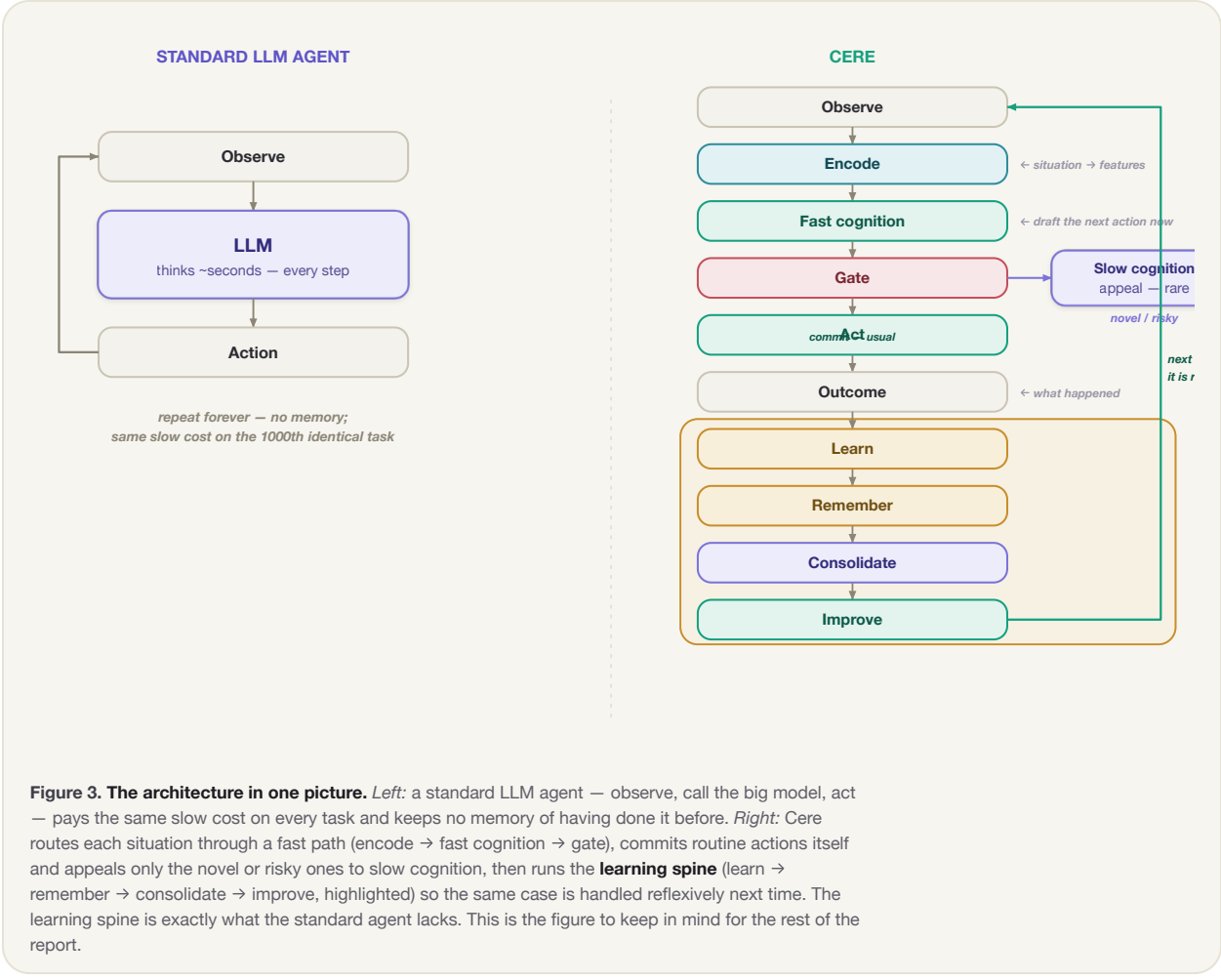


Figure 3. The architecture in one picture. *Left:* a standard LLM agent — observe, call the big model, act — pays the same slow cost on every task and keeps no memory of having done it before. *Right:* Cere routes each situation through a fast path (encode → fast cognition → gate), commits routine actions itself and appeals only the novel or risky ones to slow cognition, then runs the **learning spine** (learn → remember → consolidate → improve, highlighted) so the same case is handled reflexively next time. The learning spine is exactly what the standard agent lacks. This is the figure to keep in mind for the rest of the report.

5 What success looks like

Success is a system where:

- **routine work runs locally, instantly, and improves with use** — the thousandth invoice is filed reflexively, for free;
- **the expensive model is reserved for genuine novelty and high stakes** — you pay for the big brain exactly when you need it;

- **competence accumulates across sessions** instead of being re-bought on every call; and
- **safety is structural, not hopeful** — anything irreversible stays behind the slow model until the fast path has *earned* the right to do it, and even then there is a mechanical gate that cannot be argued with.

! Non-goals, stated up front

No hardware (software-only, for cheapness and reproducibility); **not a frozen draft model** (the online learning loop *is* the point); and **not "replace the big model"** (the big model stays — as teacher, verifier, and court of appeals).

KEY TAKEAWAYS

- Today's agents pay full price and full delay on every action — and never get faster with practice.
- Cere pairs a slow model with a tiny fast one that predicts the next action *now* and **learns** from both the slow model and the real outcome.
- Routine work migrates slow → fast; the big model is demoted to a **court of appeals** for novelty, risk, and dispute.
- Cost then tracks task **difficulty**, not request **volume** — the whole game.

PART II

The whole system on one page — then each piece, and how it grew

The Architecture

Cere was not designed all at once. It started as a two-box latency trick and grew one organ at a time, each added only after the previous one was proven. Seeing that growth is the fastest way to understand why each piece exists.

WHAT YOU WILL LEARN

- The complete architecture as one perception → action → learning cycle.
- Why Cere has two minds — a fast System 1 and a slow System 2 — and how work moves between them.
- The four organs of the fast path, the safety gate, the value and planning machinery, and the learning spine.
- How the architecture grew across six generations without ever replacing its core.

WHY THIS MATTERS

If Part I is the idea, Part II is the machine. Everything in the rest of the report is one of these boxes examined in depth, or the loop they all sit inside exercised in a harder setting. Get this map in your head and the other 180 pages are variations on it.

KEY IDEAS

- One loop: perceive, predict, verify, act, learn, remember, consolidate.
- Jurisdiction — the *moving boundary* of what the fast path may decide alone.
- Two kinds of memory: a cache that never forgets and a net that adapts.
- A sealed safety gate that the rest of the system operates *inside*.

1 The whole system on one page

Here is Cere with every major subsystem in place. Do not try to absorb it all at once — the rest of Part II is a guided tour of each box. Think of this as the map you will keep returning to.

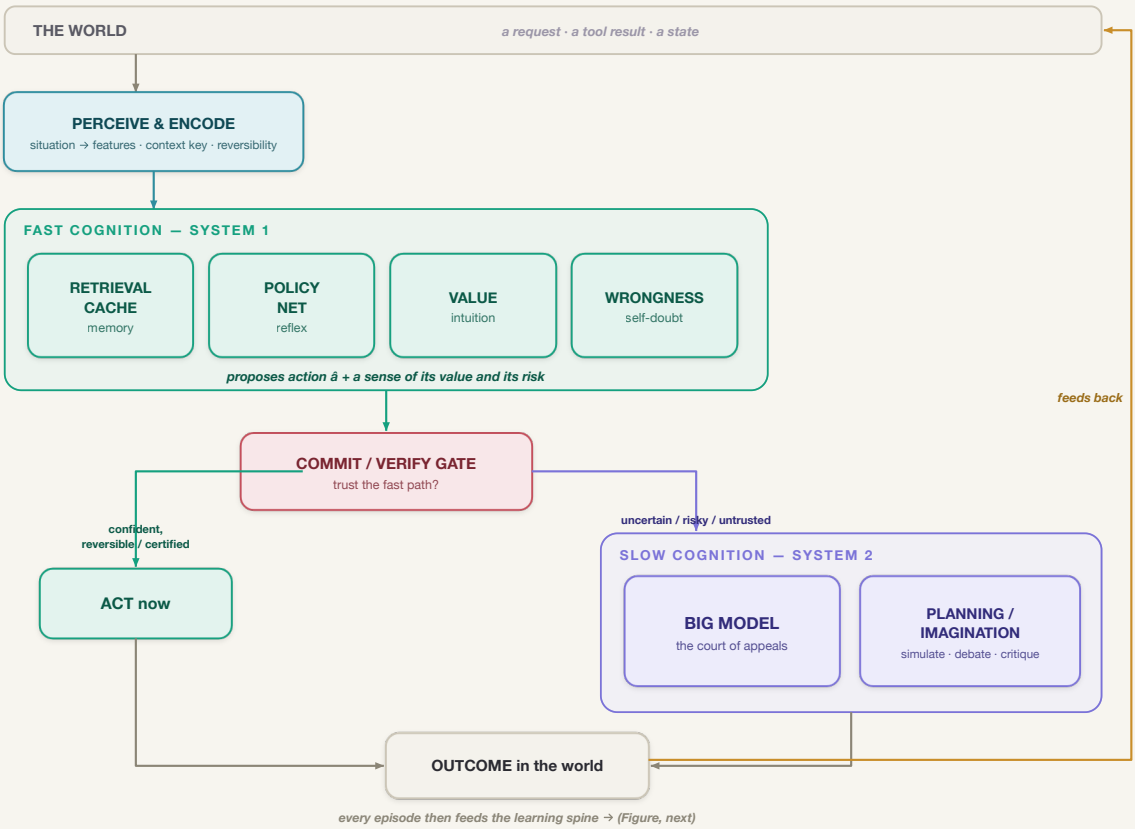


Figure 4. The Cere architecture, as one perception → action → learning cycle. The world produces a situation; Cere perceives and encodes it; the fast path (System 1) proposes an action together with a sense of how good and how risky it is; a gate decides whether to trust that or appeal to the slow path (System 2); an action is taken; the outcome feeds back. The next time a similar situation appears, it sails through the fast path.

> *That migration — slow to fast, deliberate to reflexive — is the whole machine.*

2 Two minds: System 1 and System 2

The deepest organizing idea in Cere is borrowed from psychology — Daniel Kahneman's distinction (**Kahneman, 2011**) between **System 1** (fast, automatic, intuitive: recognizing a face, catching a ball, reading a word) and **System 2** (slow, effortful, deliberate: multiplying 17 × 24, planning a trip). A healthy mind uses both, and critically, it *moves work between them*: things you do over and over migrate from effortful System 2 to automatic System 1 — the classic cognitive → associative → autonomous progression of human skill acquisition (**Fitts & Posner, 1967**). We use these as *organizing metaphors*, not claims of cognitive fidelity; the technical content is the migration mechanism, not the psychology.

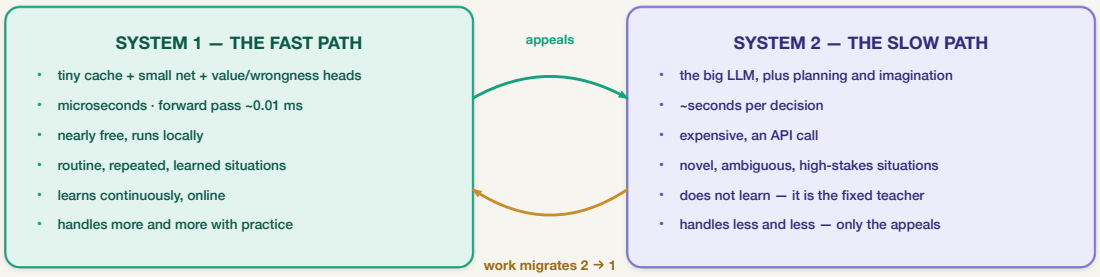


Figure 5. Cere's two minds. Routine work migrates from the slow, expensive System 2 to the fast, free System 1 with practice; the fast path appeals back to the slow one only on novelty, risk, or dispute.

◐ The boundary moves

A new employee checks every decision with their manager. A seasoned one handles almost everything alone and walks into the manager's office only for the genuinely hard calls. Nothing about the employee's *authority* changed by decree — they **earned** a wider boundary through a track record. Cere's gate is exactly this: not a fixed rule about what the fast path may do, but a moving boundary that widens as the fast path proves itself and snaps back the instant it stops being reliable.

§ Jurisdiction

the learned, moving boundary of what the fast path may decide on its own versus what must be appealed. Learning it is, in one sentence, what the entire project is about.

3 Perception and encoding

What problem exists? A tiny model cannot read a raw email or a screenshot the way a big model can. It needs the situation distilled into a compact, comparable form.

What Cere adds. An *encoder* turns each situation into three things: a **feature vector** (numbers the small net can compute on), a **context key** (a fingerprint for the retrieval cache — "have I seen this exact kind of situation before?"), and a **reversibility class** (is the action this might lead to reversible, simulated, read-only — or irreversible?). Everything downstream depends only on this encoded context, never on the raw input — which is what lets the same machinery work whether the "situation" is a ball's position, a tool call, a math problem, or a reasoning step.

4 The fast path — four small organs

Fast cognition is not one model; it is a small committee, each member cheap and each answering a different question.

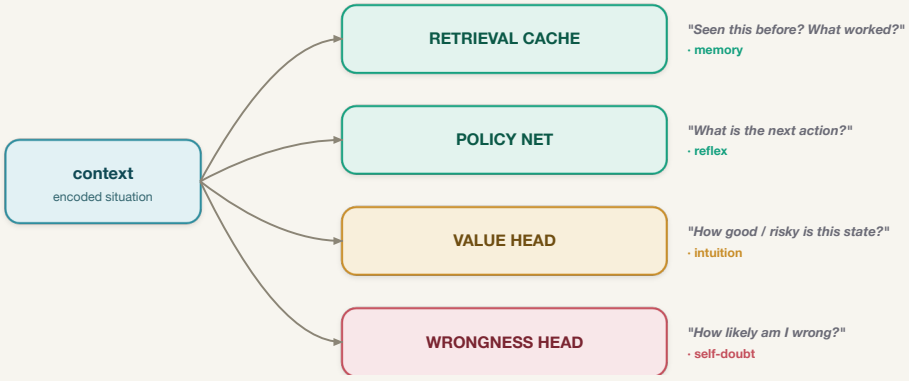


Figure 6. The four organs of the fast path. The cache and the policy net are two kinds of memory played against each other; the value and wrongness heads supply the system's intuition and its self-doubt.

- The **retrieval cache** is count-based memory: it remembers verified (context → action) pairs and never forgets them (by construction). This is the classical *retrieve-and-reuse* idea of case-based reasoning (**Aamodt & Plaza, 1994**; **Kolodner, 1993**), and the modern non-parametric, memorize-then-retrieve idea behind nearest-neighbour LMs and retrieval-augmented generation (**Khandelwal et al., 2020**; **Lewis et al., 2020**). Its weakness is that it is *count-based* — confident about anything it has seen often, even if the world has since changed.
- The **policy net** is a small trainable network. It generalizes to situations the cache has not seen exactly, and — being gradient-trained — it *adapts* when the world shifts. Its weakness is the mirror image of the cache's: it can forget.
- The **value head** is the system's *intuition* (see §6).
- The **wrongness head** is the system's *self-doubt* — it predicts the probability that the fast path's own proposal is wrong, which is what lets the gate escalate intelligently instead of on a hand-tuned threshold.

Why two kinds of memory?

The cache never forgets but never adapts; the net adapts but can forget. Keeping both, and playing them against each other, is a software echo of *complementary learning systems* theory (**McClelland et al., 1995**; **Kumaran et al., 2016**) — a fast-binding hippocampal store beside a slow-generalizing neocortex — and of the classic stability–plasticity dilemma. It is the reason the system can be simultaneously stable (old skills survive) and plastic (it tracks a changing world). One of the most reliable findings: a gradient-trained head *beats* the count-based cache exactly when the world has shifted, because the cache stays "confidently wrong" while the net notices the drop in success and updates.

5 The safety gate — the one boundary that never moves

What problem exists? A fast model that learns will sometimes be wrong. If "being wrong" can mean *sending the email to the wrong person* or *deleting the only copy of a file*, then speed is not worth it. Mistakes on irreversible actions are catastrophic.

Why the obvious fix fails. A fixed conservative threshold throws away the whole benefit (you are back to calling the big model constantly); a fixed permissive threshold is unsafe. The threshold cannot be a constant.

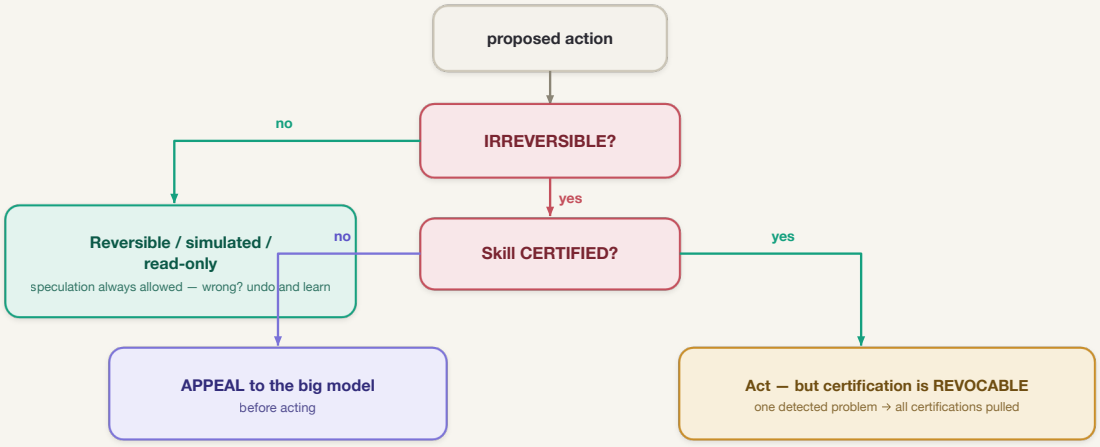


Figure 7. The safety gate splits the world by reversibility and treats the two halves completely differently. Reversible actions may always be speculated on; irreversible ones stay behind the big model until the skill is certified — and that certification is revocable the instant a problem appears.

This is the project's single most important safety property, and it holds in two forms. In the world of *actions*, the boundary is **reversibility**. In the world of *reasoning* (Part III), thinking has no irreversible step — you cannot "un-think" a thought — so the boundary flips to **verification**: a speculative answer is always safe to *attempt*, and the danger is a silently wrong answer, caught by

checking it before committing. Same gate, two faces: a *reversibility floor* for actions, a *verification floor* for thoughts.

The evidence, stated starkly

Across the experiments, this gate is tested adversarially — deliberately feeding the system situations engineered to make the fast path mis-fire on an irreversible step. In every such test we ran, the result was the same: **zero unsafe irreversible commits passed the gate**, while a pure-autonomy baseline committed every single engineered error. In one run, conservative verification caught 332 of 332 mis-fires; in another, 350 of 350 wrong answers were caught before commit. These are results on the evaluated substrates, not a theorem; what makes them credible is that the gate is *mechanical* — it does not learn, it cannot be argued with, and later arcs that let Cere modify *itself* are structurally forbidden from touching it, so its guarantee does not erode as the system grows.

6 Intuition and imagination — value and planning

Reflexes and memory let Cere reproduce what the teacher *does*. Two further organs let it learn *why* the teacher prefers some futures over others — and then act on that judgment.

The value head — intuition

The value head is your gut feeling

Before you make a chess move, you do not calculate every consequence; a trained sense tells you "this position feels strong" or "this feels like a trap." That instant, holistic judgment is what a *value function* computes. It is exactly the trick behind AlphaGo (**Silver et al., 2016**): a value network that looks at a board and estimates who is winning, without playing the game out.

Cere's value head predicts, for any situation, four numbers: how likely the task is to *succeed*, its *cost*, its *risk*, and the chance it will need *take-over* by the slow model. It is trained for free from outcomes the supervisor is already logging — no new teacher required — and it is **calibrated** (when it says 70%, it is right about 70% of the time, measured by expected calibration error (**Guo et al., 2017**)). Calibration is what makes it trustworthy enough to gate on.

Planning and imagination

Imagination is mental time-travel

Before a tricky parking maneuver, you imagine "if I cut the wheel now, I'll clip the curb; if I wait two feet, I'll line up." You are running a fast internal simulation and judging the outcomes — searching imagined futures and picking the best one before moving a muscle.

Cere learns a **world model** (Ha & Schmidhuber, 2018; Hafner et al., 2020) (a fast predictor of "if I do X, what state results?"), imagines several futures, scores each with the value head, and expands the promising ones — *imagine → score → explore*. This is the same family of idea as Monte-Carlo tree search (Coulom, 2006; Kocsis & Szepesvári, 2006) in AlphaGo, and as planning with a *learned* model in MuZero (Schrittwieser et al., 2020) — but guided by learned intuition rather than blind enumeration.

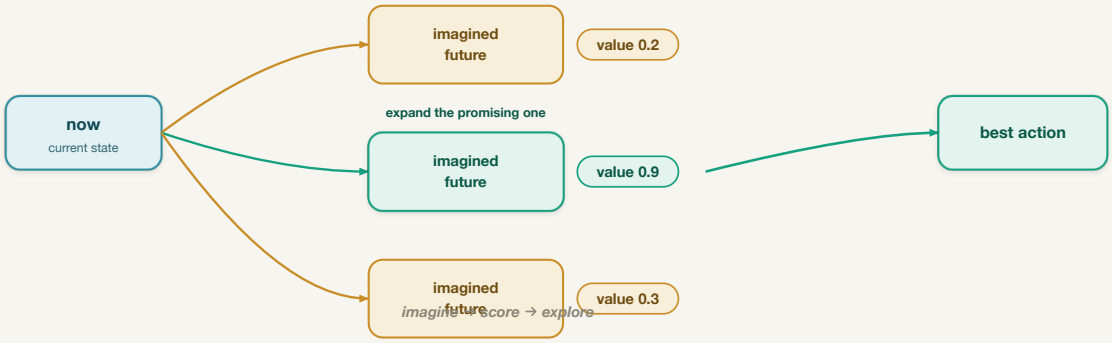


Figure 8. Guided imagination: roll futures forward in a learned world model, score each with the value head, and expand only the promising branches within a fixed budget. When the per-step guess is noisy, this wins overwhelmingly over both greedy action and blind search at the same compute.

7 The learning spine — learn, remember, consolidate, improve

Everything above would be a clever-but-static system without the spine that makes it *get better*.

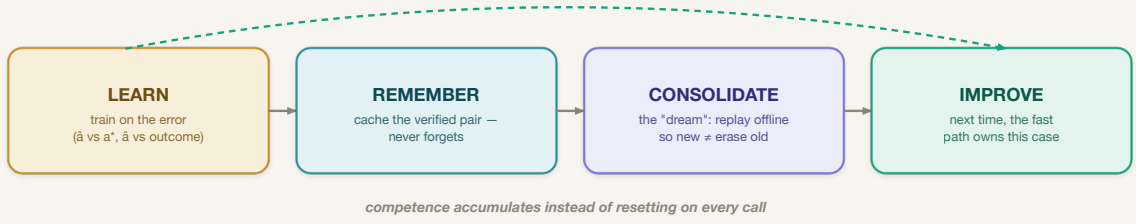


Figure 9. The learning spine. Four steps turn a single episode into a permanent improvement, so the system's competence accumulates instead of resetting on every call.

◐ Why we sleep on it

Your brain consolidates the day's learning during sleep, replaying experiences so they settle into long-term memory without erasing what you already knew. The dream pass is the same idea: learn fast and a little carelessly while awake; replay and tidy up while idle, so today's skill does not cost yesterday's.

Stated technically: training a network purely online causes **catastrophic forgetting** (McCloskey & Cohen, 1989; French, 1999) — gradients for a new skill overwrite the weights that encoded an earlier one. The dream pass is **experience replay** (rehearsal) (Lin, 1992; Robins, 1995): interleaving stored past episodes with new ones during idle batch training — the same anti-forgetting mechanism that stabilizes deep Q-networks (Mnih et al., 2015), and a complement to regularization approaches such as elastic weight consolidation (Kirkpatrick et al., 2017) (for the broader setting, see continual-learning surveys, Parisi et al., 2019).

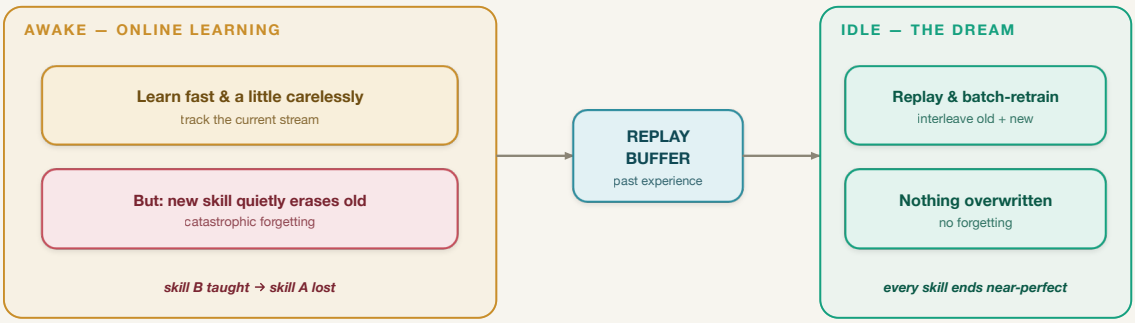


Figure 10. The dream. A model that only learns online forgets — teach it skill B and it quietly erases skill A. During idle time Cere replays a buffer of past experience and batch-retrains, interleaving old and new so nothing is overwritten.

The evidence

Run the system online-only and a skill peaks at 0.80 then collapses to 0.40 once a later skill is taught. Add the dream pass and, in this experiment, every skill ends near-perfect, the earliest recovering to 1.00. Forgetting is real, and on this substrate the dream pass is what defeats it.

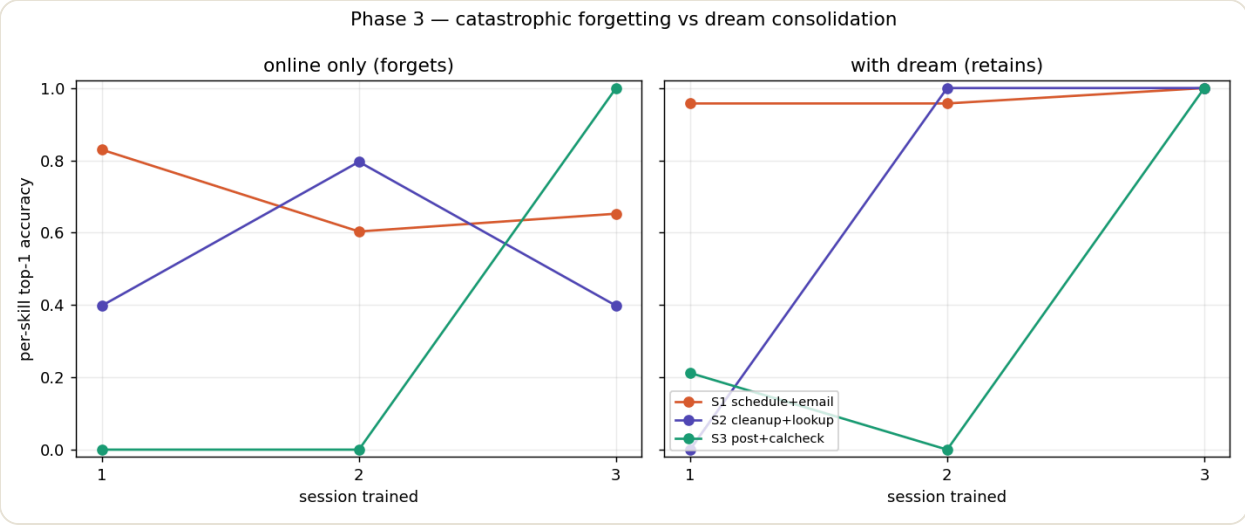


Figure 11. Online-only learning forgets: an early skill peaks and then collapses when a later one is taught (the dip). With the dream pass, every skill is retained. Forgetting is real, and offline replay is what defeats it.

8 How the architecture grew

Cere **grew one organ at a time**, each added only after the previous one was proven. Every box earned its place by solving a problem the previous generation could not.

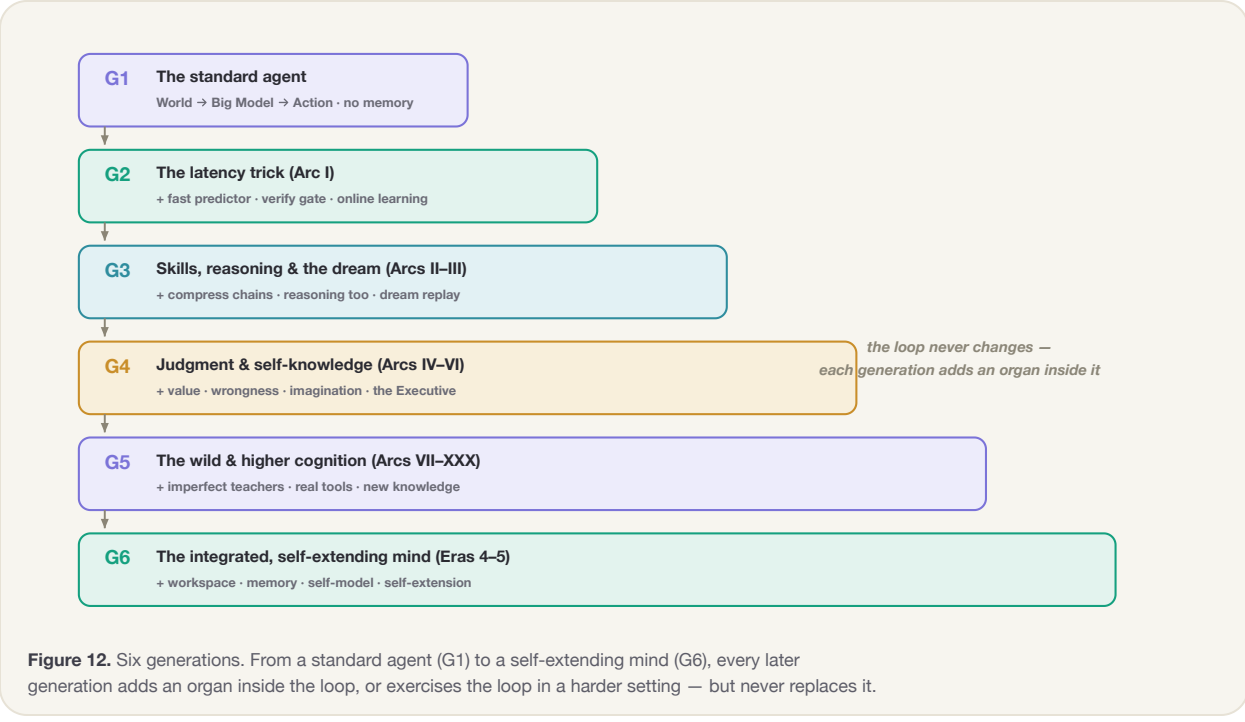


Figure 12. Six generations. From a standard agent (G1) to a self-extending mind (G6), every later generation adds an organ inside the loop, or exercises the loop in a harder setting — but never replaces it.

Notice what *does not change* from Generation 2 onward: the loop — perceive, predict, verify, act, learn, remember, consolidate — is identical. This is the

single most important structural fact about Cere, and Part V is devoted to it:
one learning loop, elaborated 200 times.

Figure 13. Action migration, the unit of progress. A context that is appealed the first time is learned over the next several, and handled reflexively by the hundredth — the boundary of the fast path's jurisdiction widening as it earns trust.

9 The vocabulary: faculties, stages, and systems

As the architecture grew to 39 arcs, the project adopted a three-way distinction to keep the map honest.

Class	What it is	Examples
Faculty	*new learnable machinery* — an organ the system learns to do	reflexive prediction · value · causal understanding · theory of mind
Stage	*a harder setting* exercising existing faculties — no new mechanism	survive the wild · do office work · act on initiative · a collective
System	*connective & regulatory tissue* that makes the faculties one mind	global workspace · attention · memory indexing · the self-model

Table 1. The test for any addition: is it a new learnable capability (faculty), a new setting for existing ones (stage), or new tissue that integrates existing faculties (system)? Keeping these straight is what lets a 200-phase roadmap stay legible.

KEY TAKEAWAYS

- The whole system is one loop: perceive → predict → verify → act → outcome → learn → consolidate.
- Two minds — a fast, learning System 1 and a slow, fixed System 2 — with work migrating 2 → 1 and appeals going 1 → 2.
- The fast path is a committee of four cheap organs; the gate is a sealed, reversibility-aware safety boundary the rest of the system runs inside.
- Value and imagination add judgment; the spine adds accumulation; six generations of growth never replaced the core loop.

PART III

Each cognitive “organ”: the problem it solves, how it works, the proof

The Faculties

Every faculty below is, underneath, the same loop from Part II pointed at a new kind of content. Once you have read the first one in full, the rest are variations on a theme — which is exactly the project's central claim.

WHAT YOU WILL LEARN

- The nine core faculties — built and proven most rigorously, in Era 1 — in full.
- For each: purpose, the problem, why earlier approaches failed, the mechanism, a worked example, and the experimental evidence.
- How the faculties interact, and where each one's honest limits are.
- A grouped tour of the higher-order faculties that reuse the same machinery.

WHY THIS MATTERS

Part II showed the machine; Part III opens it up. These are the learnable organs — reflex, skill, reasoning, value, planning, metacognition, robust learning, open-ended judgment, and the executive that ties them together. Each was given its own chance to fail on the smallest substrate that could express it.

KEY IDEAS

- Purpose · Problem · Why fixes fail · Seed · Mechanism · Example · Evidence · Limits — the same rhythm throughout.
- Every faculty is calibrated, learned online, and keeps its irreversible steps behind the sealed gate.
- The transfer library for skills is literally the same code reused for reasoning.
- Everything converges on the Executive — the big model as court of appeals.

How to read each faculty

Purpose = the capability. *Problem* = why you need it. *Why obvious fixes fail* = what makes it non-trivial. *Seed* = the loose biological inspiration (a design hint, never a claim of replication). *Mechanism* = inputs → computation → outputs. *Example* = a concrete picture. *Evidence* = the experiment and its key numbers. *Interacts* = which other organs it talks to. *Limits* = the honest edge.

CORE FACULTY 1 · ARC I · PHASES 1–4

F1 Forward Prediction & Reflex

Predict the near-future situation faster than real time and emit the next routine action *now*, hiding the slow model's latency.

THE PROBLEM

The slow model's decision arrives too late to be useful at speed; by the time it answers, the situation has moved on.

WHY OBVIOUS FIXES FAIL

Just using a smaller model loses competence. Extrapolating with a fixed rule (“assume it keeps moving straight”) works for trivial dynamics and collapses on anything curved or branching. You need a predictor that *learns the specific dynamics it faces*.

Biological seed

The cerebellum's forward models (**Wolpert et al., 1998**; **Miall et al., 1993**); Kahneman's System 1 (**Kahneman, 2011**). The draft-then-verify shape is speculative decoding (**Leviathan et al., 2023**) with a *learning* draft.

MECHANISM

A tiny trainable net takes the encoded situation — delayed by the latency budget — and predicts the present-and-next state and the next action; a gate decides whether to commit. The proposed action $\hat{a}$ is committed immediately when confident and reversible, and trained online from the slow model's eventual decision and the real outcome.

WORKED EXAMPLE

A 2-D interception task: a target moves on a curved, drifting path, and every policy sees the world only as it was a moment ago. The naive "act on the stale observation" policy fails on the curves; the learned predictor steers to where the target *will be*.

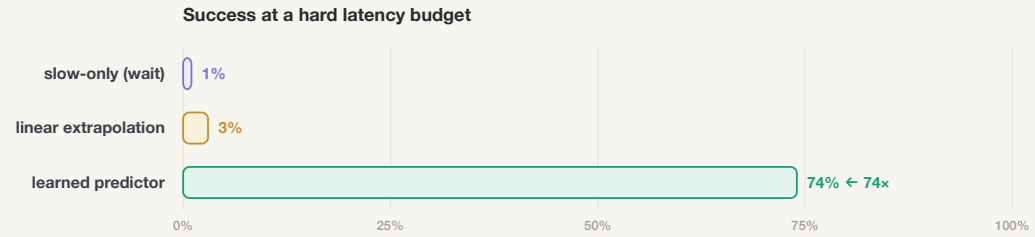


Figure 14. At a latency where waiting essentially never works, the learned predictor succeeds three-quarters of the time — the headline of Arc I.

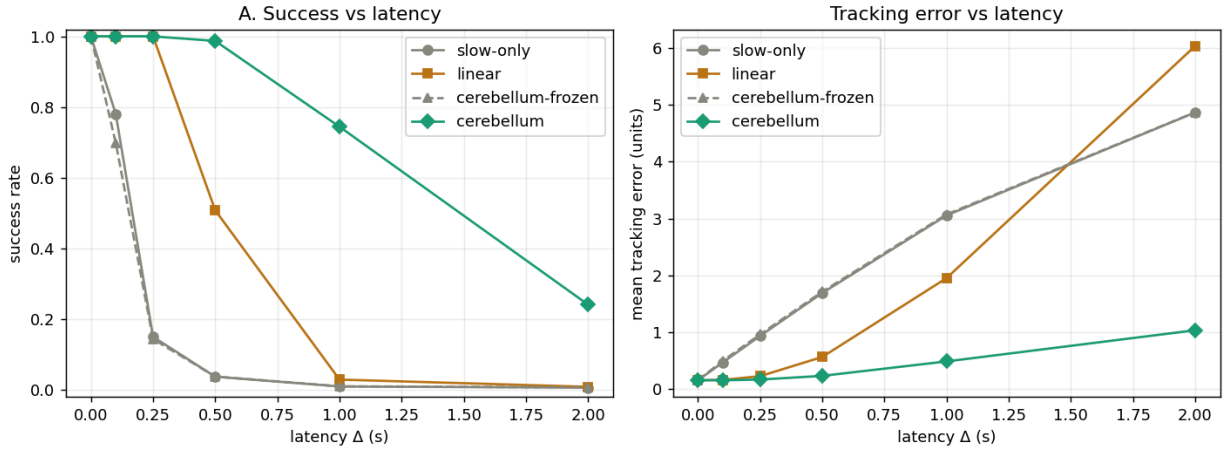


Figure 15. Success vs latency. As the latency budget grows, baselines collapse while the learned cerebellum holds — it acts on a prediction, not a stale observation.

What the experiment showed

At a latency where the slow-only baseline succeeds 1% of the time and a linear extrapolator 3%, the **trained predictor reaches 74%**, learning it in about 20 episodes (0.09 → 0.72). Its forward pass is **0.011 ms against a 1000 ms budget — roughly 90,000× of headroom**. On tool-call workflows the slow-call rate fell **0.72 → 0.25** (the irreversibility floor) while task success stayed **1.00**, saving ~3,700 seconds of slow-model time over 1,500 tasks.

INTERACTS WITH

Feeds the gate (§II.5); its errors train via the learning spine; later wrapped by Skills (F2) and judged by Value (F4).

LIMITS & FUTURE

Far-horizon prediction has a real ceiling — past a certain delay the future is genuinely unpredictable and the right move is to *bound the horizon*. The minimal environment is deterministic-given-context, so the cache is perfectly calibrated; a noisier world would stress calibration and is the top open extension.

CORE FACULTY 2 · ARC II · PHASES 5–8

F2 Skill Consolidation

Compress a *repeated chain* of decisions into a single reusable **skill**, so a routine multi-step task becomes one reflexive invocation — and migrate that skill from slow to fast safely.

THE PROBLEM

Predicting one action at a time is still a decision per step. A competent assistant does not re-derive "search → select → fill → pay" every time they book travel; they execute a known routine.

WHY OBVIOUS FIXES FAIL

Hard-coding macros is brittle (the world varies) and unsafe (a macro that includes "pay" must not fire blindly). The skill must be *mined from verified experience* and must keep its irreversible steps gated until trusted.

Biological seed

Basal ganglia (procedural memory / habit formation); the cerebellar automation of a practiced skill.

MECHANISM

Mine recurring chains keyed on (task type, what's been revealed, step) — temporal abstraction in the sense of options and feudal HRL (**Sutton et al., 1999**; **Dayan & Hinton, 1993**); when one fires, it proposes the *whole remaining chain* in one decision, with any irreversible step inside it still verified until the skill earns autonomous status. A shared **component library** lets a skill learned in one domain transfer to another (in the spirit of Voyager's growing skill library for an LLM agent (**Wang et al., 2023**), but mined from *verified* experience and kept behind the gate); idle-time **practice** rehearses weak skills; a **society of specialists** routes different skill families to different small heads.

WORKED EXAMPLE

"Book a flight" and "book a hotel" both share *search → select → fill → pay*. Learn that backbone once; reuse it for free in both, and in the next booking-shaped task you have never seen.

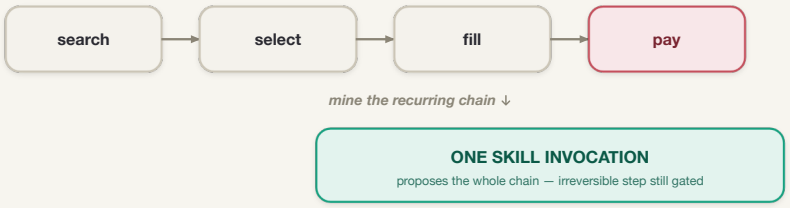


Figure 16. Skill consolidation: a verified chain of steps is mined and replaced by a single invocation — with the irreversible step ("pay") still gated until the skill is certified.

Figure 17. Decisions per task fall as repeated chains are consolidated into skills, while task success is held at 1.00.

What the experiment showed

Decisions per task fell **3.30 → 1.80 (−45%)** with success held **1.00**. Transfer: a held-out target reached competence with fewer slow calls (308 → 295) by reusing the booking backbone. Practice lifted weak-skill accuracy **0.55 → 0.91** while a uniform-practice control stayed at 0.55. A society of specialists beat a single "monolith" **0.48 → 0.75**, and breaking one specialist left the others untouched. **Safety:** on 332 out-of-distribution tasks with a perturbed irreversible step, conservative verification caught **332/332**; pure autonomy executed all 332 wrong.

INTERACTS WITH Sits on top of F1's predictor; shares the dream pass (anti-forgetting); its transfer library is *literally the same code* later reused for reasoning (F3).

LIMITS & FUTURE Validated as isolated mechanisms; cross-skill sharing across a *society* and a composite-task handoff protocol are demonstrated only partially. Folding skills into one integrated runtime is open work.

CORE FACULTY 3 · ARC III · PHASES 9–13 · THE PIVOT

F3 Reasoning Consolidation

Apply the *exact same* consolidation machinery to **reasoning itself** — learn reusable "thought-skills" instead of re-deriving every chain of reasoning from scratch.

WHY IT MATTERS This is the hinge of the whole project. The headline claim changes here, from "a small model can learn a big model's **actions**" to "a small model can learn and reuse a big model's **actions and reasoning**" — what the project calls **speculative cognition**.

THE PROBLEM Arc II made routine *actions* reflexive. But most of the slow model's expense is *thinking*, not acting. Can routine *reasoning* become reflexive too?

WHY OBVIOUS FIXES FAIL You cannot cache *answers* — the next problem has different numbers. You must cache the *procedure*: the shape of the reasoning (a consolidated chain-of-thought (**Wei et al., 2022**)), not its output — and distil it into the fast path (**Hinton et al., 2015**).

Biological seed

Cortical schema formation; the proceduralization of deliberate reasoning into intuition (System 2 → System 1) — the way a mathematician "sees" a solution a student would grind out.

MECHANISM Mine recurring reasoning chains (`identify knowns → derive → apply rule → verify`) into a single thought-skill, keyed on (family, what's revealed, step). The proposed answer is checked by **answer-verification** before commit — because thinking has no irreversible step. The whole Arc II toolkit ports over: transfer, idle dreaming, specialist societies, one unified runtime.

WORKED EXAMPLE Solving `3x + 6 = 18` and solving `5y − 2 = 13` are different problems with the same *shape*. Learn the shape once; the second is reflexive.

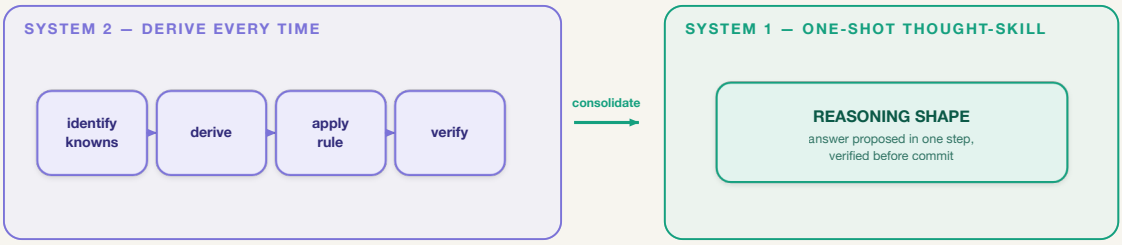


Figure 18. The pivot: the identical consolidation machinery that made actions reflexive makes reasoning reflexive. Thinking has no irreversible step, so the gate flips from reversibility to answer-verification.

Figure 19. Reasoning steps per problem fall by 62% as thought-skills consolidate — the same migration curve as actions, now over reasoning, at held accuracy.

What the experiment showed

Reasoning steps per problem fell **3.96 → 1.50 (−62%)** with answer accuracy held **1.00** — the same curve as actions, now over thought. The transfer library was the *unchanged Phase 6 code*, only the alphabet swapped from tools to reasoning ops. A society of specialist thinkers beat a monolith 0.64 → 0.88. The unified runtime ran action *and* reasoning over a 960-item lifetime: cost per task **0.64 → 0.37**, 1,708 slow calls saved, quality 1.00, nothing forgotten. **Safety:** 350/350 wrong answers caught before commit.

That the identical library works for both is itself the result

When the same transfer code, byte-for-byte unchanged, consolidates reasoning exactly as it consolidated tool use, "alphabet-agnostic" stops being a claim and becomes an observation.

INTERACTS WITH Re-uses F2's library verbatim; shares the dream pass; its verification gate is the reasoning-world twin of the reversibility gate.

LIMITS & FUTURE The reasoning families are checkable by construction (they ship exact verifiers), which is what makes the verification gate free; open-ended reasoning *without* a cheap checker is the hardest extension (taken up by F8).

CORE FACULTY 4 · ARC IV · PHASES 14, 16

F4 Intuitive Valuation

Estimate, instantly and without rolling anything out, how *good / costly / risky* a situation is — and keep that estimate honest by correcting it against what actually happened.

THE PROBLEM Every prior faculty learned a *policy* (what to do next). To decide *whether to trust* an action, or which of several to pick, you need the missing half: a sense of *value*.

WHY OBVIOUS FIXES FAIL You could estimate value by simulating outcomes — but that is slow, exactly what we are avoiding. And a value estimate that is *miscalibrated* (says 90% when it means 60%) is worse than none, because the gate will trust it.

Biological seed

Dopaminergic reward prediction; orbitofrontal subjective value; the cerebellar error loop for recalibration; AlphaGo's value network (**Silver et al., 2016**).

MECHANISM A small head predicts success / cost / risk / take-over, trained online from outcomes the supervisor already logs — *no new teacher*. The companion mechanism closes the loop: compare every prediction to the realised outcome and train on the gap, so calibration is *maintained, not granted*.

WORKED EXAMPLE Before attempting a task, the value head says "this feels like a 0.3-risk situation" — and because it is calibrated, the gate can act on that number directly.

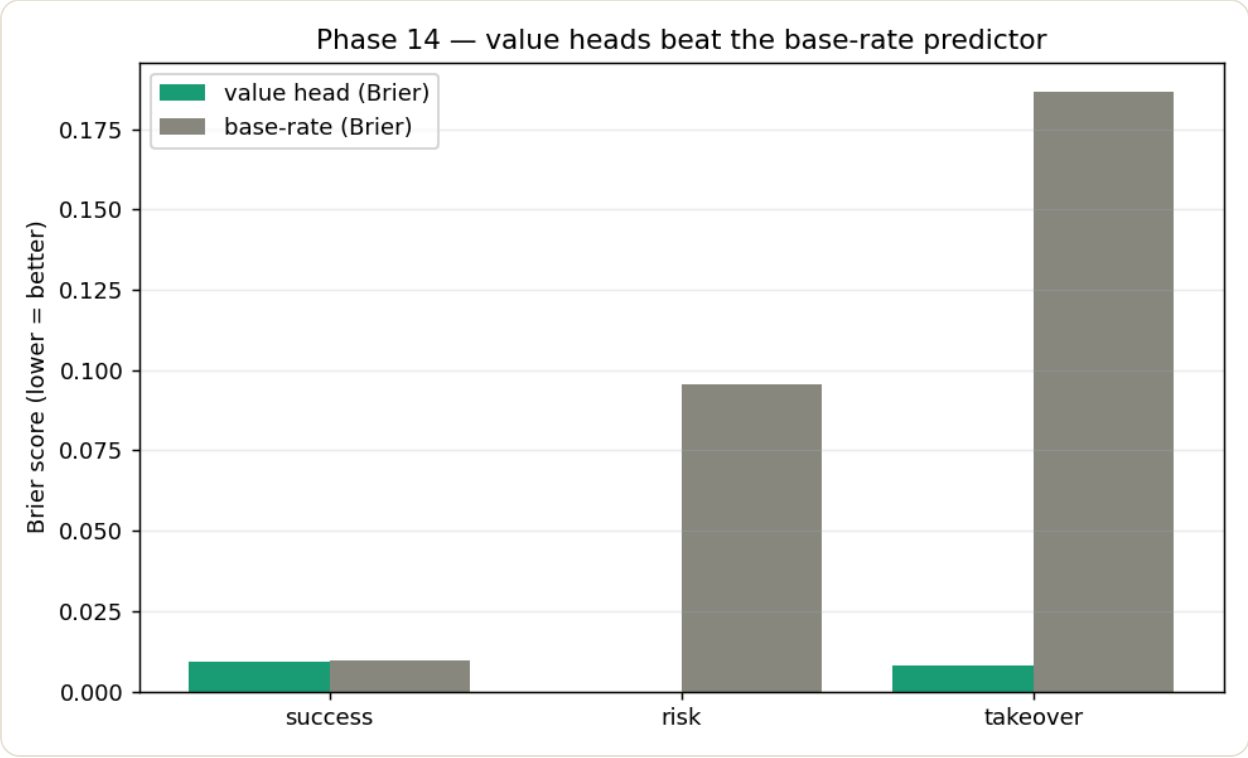


Figure 20. The value head is calibrated: predicted probabilities match realised frequencies, which is what makes it safe to gate on.

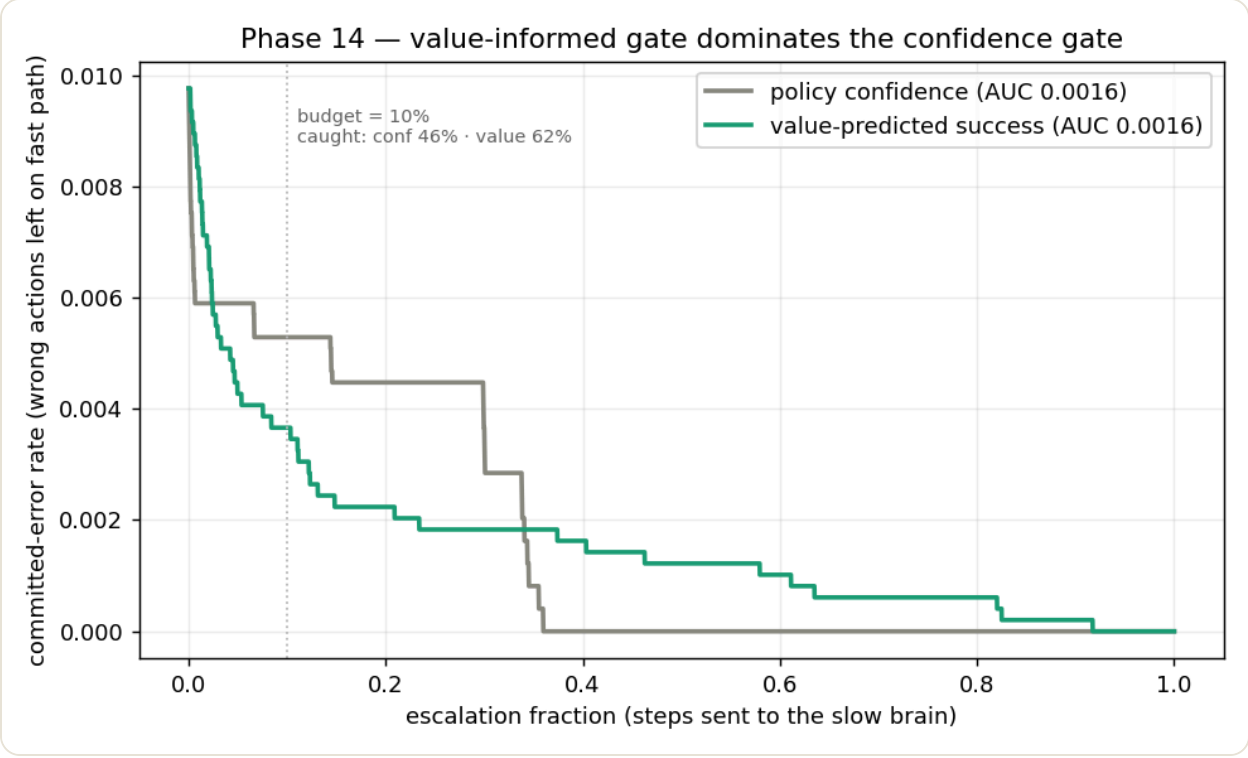


Figure 21. A calibrated value head catches more committed errors at the same escalation budget than raw policy confidence — because it tracks distribution shift.

What the experiment showed

Calibrated to within a few percent (expected calibration error ≈ 0.002). A **value-informed gate caught 62% of committed errors at a 10% escalation budget, versus 46% for policy confidence** — because the gradient-trained value head tracks a distribution shift the count-based cache stays confidently wrong about. The forward pass is 6.6 microseconds. Through a deliberate shift, an online error-loop estimator recovered (calibration error 0.33 → 0.09) where a frozen one did not, which then let a fixed gate catch **97.6%** of post-shift errors.

INTERACTS WITH — Gates with F1; scores imagined futures for F5; its features feed the wrongness head (F6).

LIMITS & FUTURE — The value-add only *shows up* under non-stationarity — on a perfectly stationary stream the policy converges to perfect and there are no errors left for value to catch. Folding explicit value recalibration into the unified pipeline is open.

CORE FACULTY 5 · ARC IV P15 + ARC IX P33–37

F5 Guided Imagination & Deep Planning

Search over *imagined* futures to choose well — far ahead, and at bounded cost — instead of acting greedily one step at a time.

THE PROBLEM — A greedy chain is fragile: if each step is only ~55% reliable, five steps deep you are almost surely off the rails. And naively imagining every branch explodes exponentially with depth.

WHY OBVIOUS FIXES FAIL — Blind search wastes its budget on hopeless branches. Flat rollouts collapse with depth as small errors compound. You need *guidance* (where to look) and *abstraction* (look at concepts, not every detail).

Biological seed

Hippocampal replay / prospection ("mental time travel"); prefrontal hierarchical control — formalized as options and feudal HRL (**Sutton et al., 1999**; **Dayan & Hinton, 1993**); Monte-Carlo Tree Search (**Coulom, 2006**; **Kocsis & Szepesvári, 2006**).

MECHANISM — *imagine* → *score* → *explore* — roll futures forward in a learned world model, rank with the value head, expand the promising ones within a fixed budget. Deep-planning add-ons: **hierarchy** (plan over subgoals), **abstract imagination** (predict the horizon-k concept directly), **branch pruning**, **counterfactuals**, and **long-horizon credit**.

WORKED EXAMPLE

Planning a multi-leg trip, you reason over *legs* ("fly to a hub, then drive"), not over every minute — and you skip obviously-bad itineraries without pricing them out.

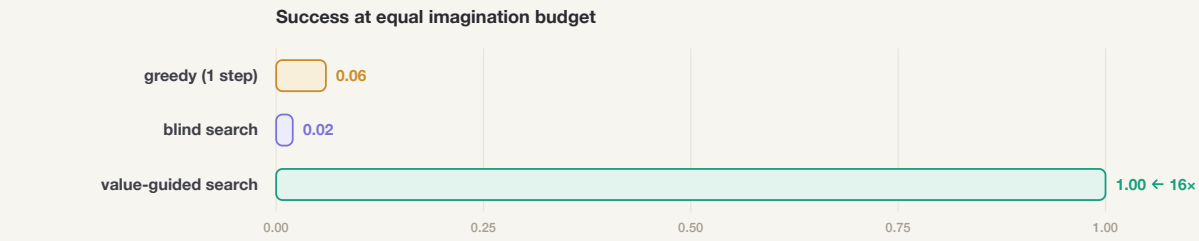


Figure 22. Same compute, three strategies. Guided imagination wins overwhelmingly.

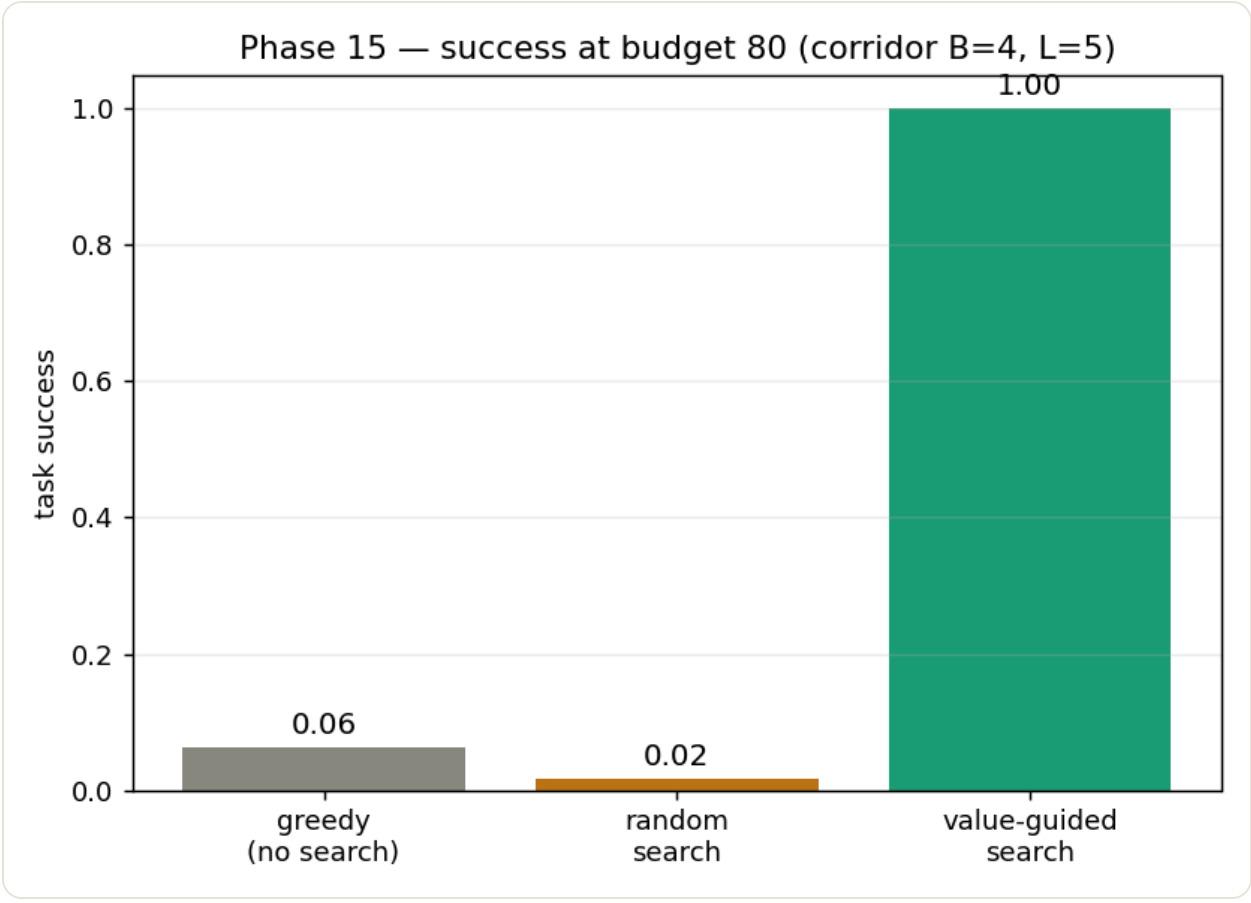


Figure 23. Value-guided search versus greedy and blind search at an equal imagination budget — guidance is what makes imagination pay.

What the experiment showed

At an equal imagination budget, **value-guided search scored 1.00 versus 0.02 for random and 0.06 for greedy**. Hierarchy turned exponential into additive: at depth 9, **1.00 versus 0.005 flat**, with the reliable horizon growing from 2 to 8. Predicting the concept directly was **0.92 accurate at horizon 16 versus 0.79**, using *1 model evaluation instead of 36*. A learned pruner held full quality at width 4 of 6 (catastrophic-prune 0.037 vs 0.441). Long-horizon credit lifted the policy to **0.945 vs 0.197**.

INTERACTS WITH — Driven by F4's value; its world model is shared with reasoning; feeds the executive's decisions.

LIMITS & FUTURE — The world model in the search experiment is deliberately near-perfect to isolate *search* from *model error*; a noisy learned model is the harder, integrated case.

CORE FACULTY 6 · ARC V · PHASES 17–19

F6 Self-Knowledge & Control (Metacognition)

Reason about its *own* cognition: predict when it is about to be wrong, decide how long to think, and reuse the cognitive strategies that keep working.

THE PROBLEM — The gate had been using a hand-tuned confidence threshold. But "confidence" and "correctness" are not the same thing, and a fixed thinking budget wastes effort on easy problems while starving hard ones.

WHY OBVIOUS FIXES FAIL — Raw model confidence is systematically fooled — modern networks are poorly calibrated out of the box (**Guo et al., 2017**), and a count-based cache is *most* confident exactly where it has thin, stale data. You need a *learned* estimate of your own error (the selective-prediction problem (**Geifman & El-Yaniv, 2017**; **El-Yaniv & Wiener, 2010**)), and a *learned* allocation of effort.

Biological seed

Anterior cingulate / metacognitive prefrontal cortex — confidence, error monitoring, effort allocation.

MECHANISM

A **wrongness head** predicts `P(my own proposal is wrong)` from features the fast path already has, trained from realised correctness. A **budget controller** spends *just enough* repeated attempts to clear a confidence target. A **strategy** layer consolidates named, abstract principles ("when underspecified, gather information first") one rung above concrete skills.

WORKED EXAMPLE

Faced with an unfamiliar, malformed request, the wrongness head spikes — "I don't know this" — and the system appeals, rather than confidently doing the wrong thing.

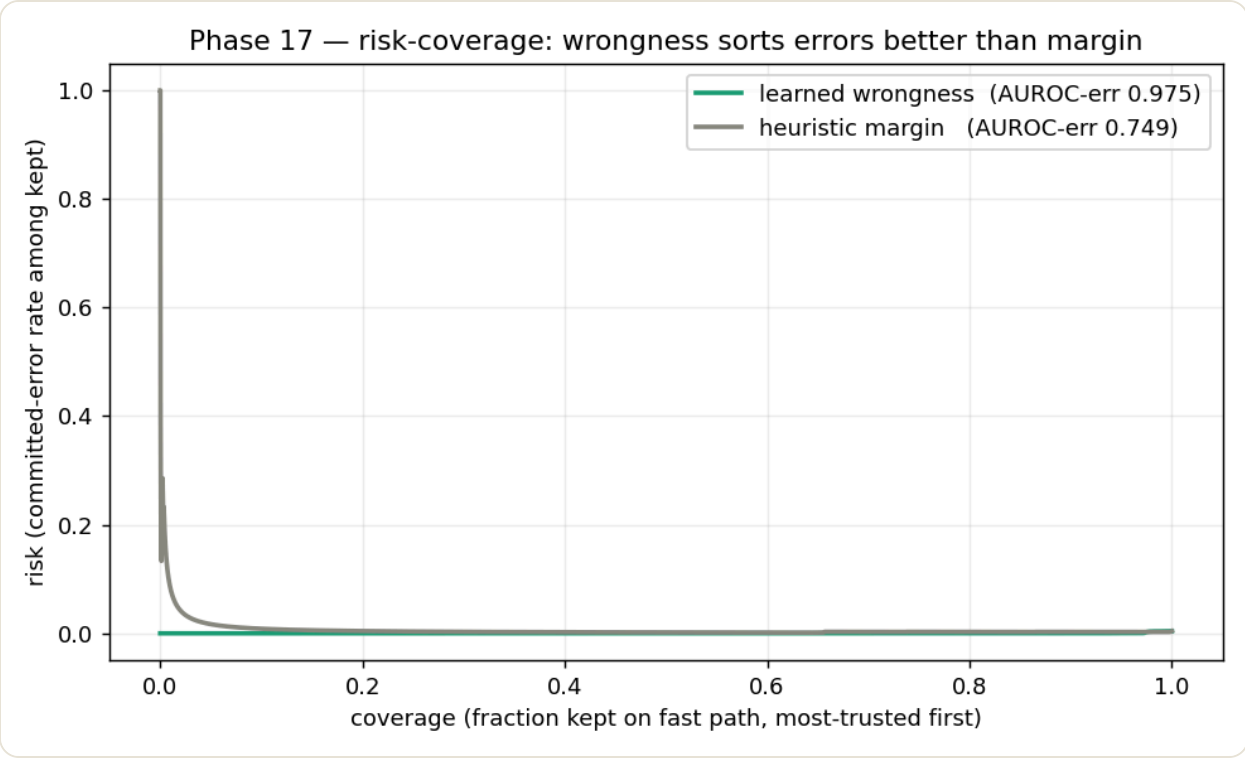


Figure 24. Risk–coverage: the learned wrongness head retains far more accuracy at any coverage than a hand-tuned confidence margin.

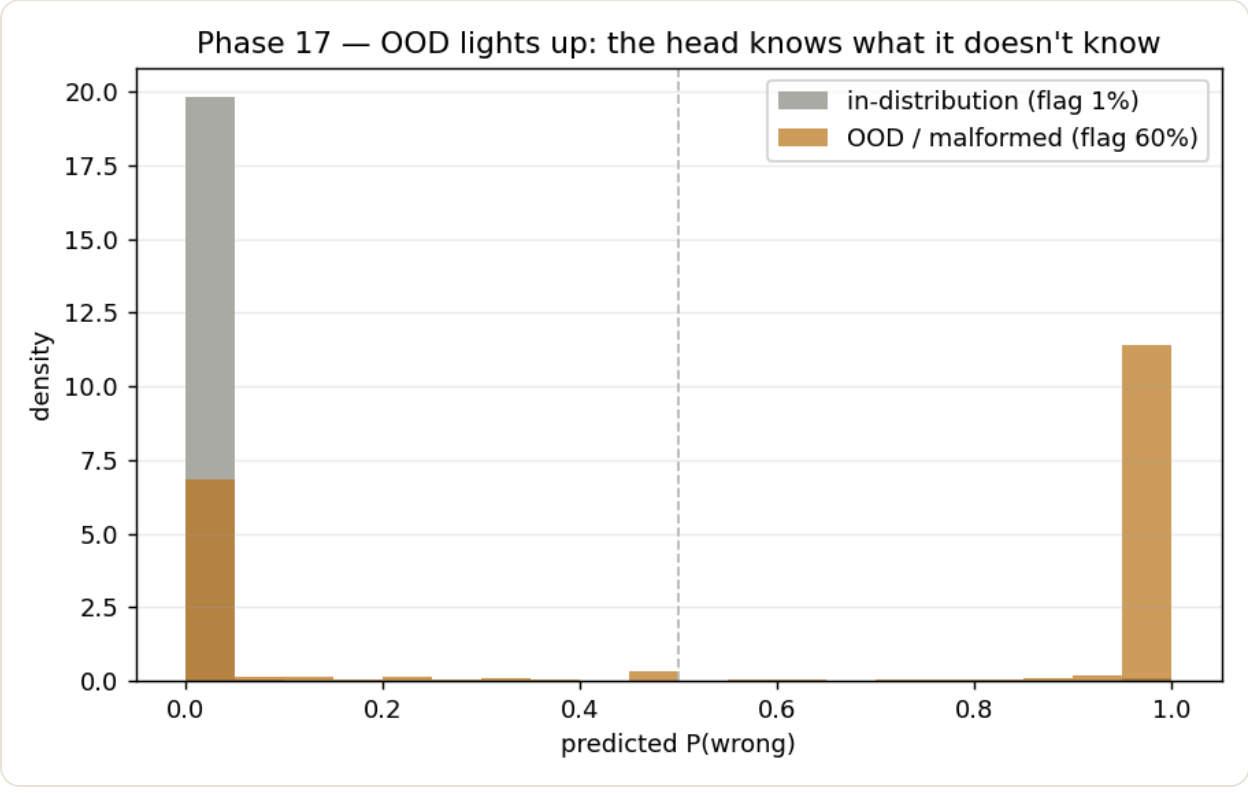


Figure 25. It knows what it doesn't know: on out-of-distribution, malformed inputs the wrongness signal spikes, driving an appeal instead of a confident error.

What the experiment showed

The learned wrongness head reached **AUROC-of-error 0.983 versus 0.749** for the hand-tuned margin; at a 10% escalation budget it caught **98% of committed errors versus 46%**. On out-of-distribution states it flagged high wrongness **60% of the time versus 1% in-distribution** — *it knows what it doesn't know*. The adaptive budget Pareto-dominated every fixed budget; at equal mean compute it solved the hardest slice **0.899 versus 0.677**. One named strategy formed and was reused across all 16 task families. Overheads: 1–7 microseconds.

INTERACTS WITH Replaces the hand-tuned threshold in the gate; consumes F4's value; its confidence signal was confirmed on *real models* (a distilled 0.5B model's own token confidence separated its right answers from its wrong ones).

LIMITS & FUTURE Strategies are validated on a synthetic family set; richer, conflicting strategy libraries are future work.

CORE FACULTY 7 · ARC VII · PHASES 24–27

F7 Robust Distillation from Imperfect Teachers

Learn well even when the teacher is *not* reliable — when teachers disagree, vary, drift, and are sometimes systematically wrong — and re-

cover the *latent principle* rather than copying surface behaviour, so the student can *surpass* the teacher and survive its replacement.

THE PROBLEM

Every earlier arc assumed a mostly-correct, checkable teacher. Real teachers (including real LLMs) are none of those things consistently.

WHY OBVIOUS FIXES FAIL

Plain imitation ("behaviour cloning" (**Pomerleau, 1991**; **Ross et al., 2011**)) bakes in the teacher's mistakes, caps the student at the teacher's level, and shatters when the teacher is swapped. Majority vote fails when the unreliable outnumber the reliable.

Biological seed

Social learning with source credibility — learning the *rule* behind a demonstration, not the demonstration itself.

MECHANISM

Detect **disagreement** (vote entropy, calibrated to majority-correctness). Estimate per-teacher **reliability with no answer key** (Dawid–Skene EM (**Dawid & Skene, 1979**) — infer who is trustworthy purely from agreement patterns). Detect **drift** (a change-detector robust to the very corruption it watches for). And **infer the principle** — fit a rule-form model that averages out each teacher's noise and style.

WORKED EXAMPLE

Three reviewers grade an essay; two share a quirk, one is sharp but contrarian. Don't average blindly — infer who is reliable from how they agree and disagree, then weight accordingly.

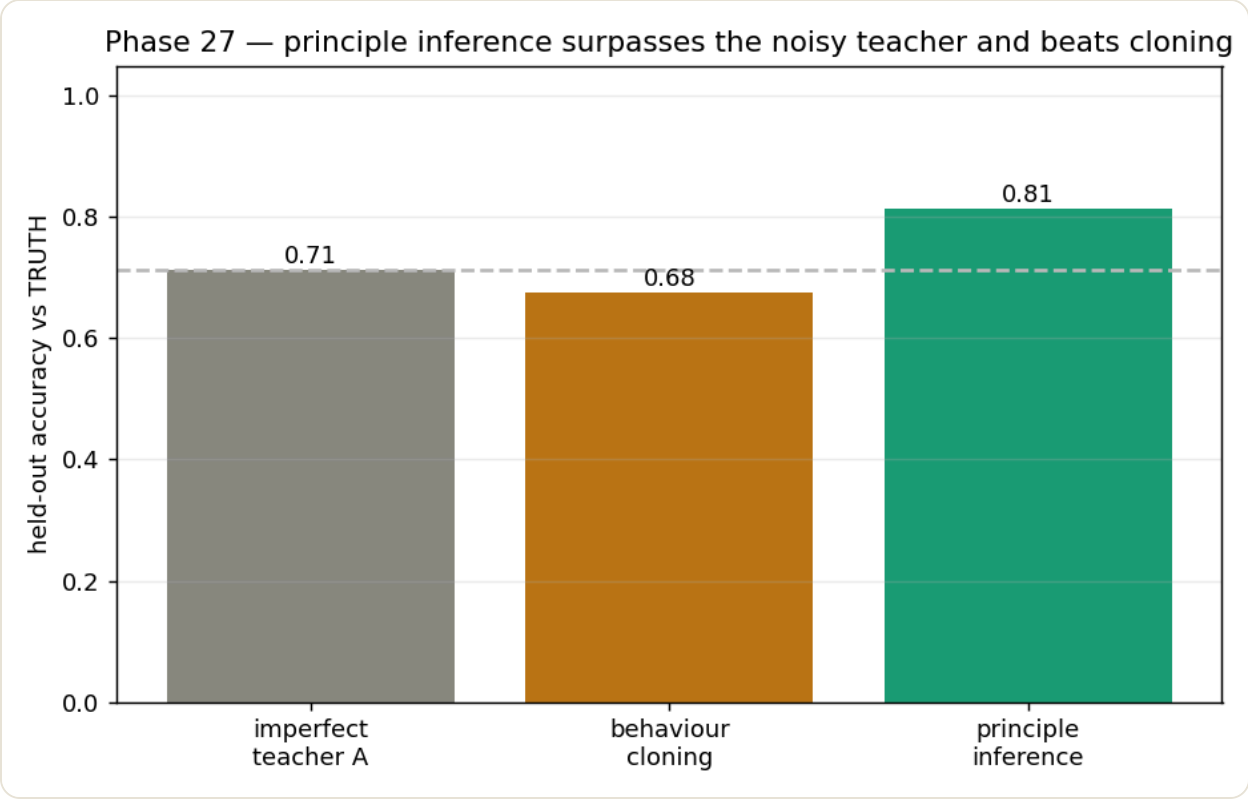


Figure 26. Principle inference exceeds the teacher's own accuracy and beats behaviour cloning — and, because it captured the rule, it survives a teacher swap with no re-learning.

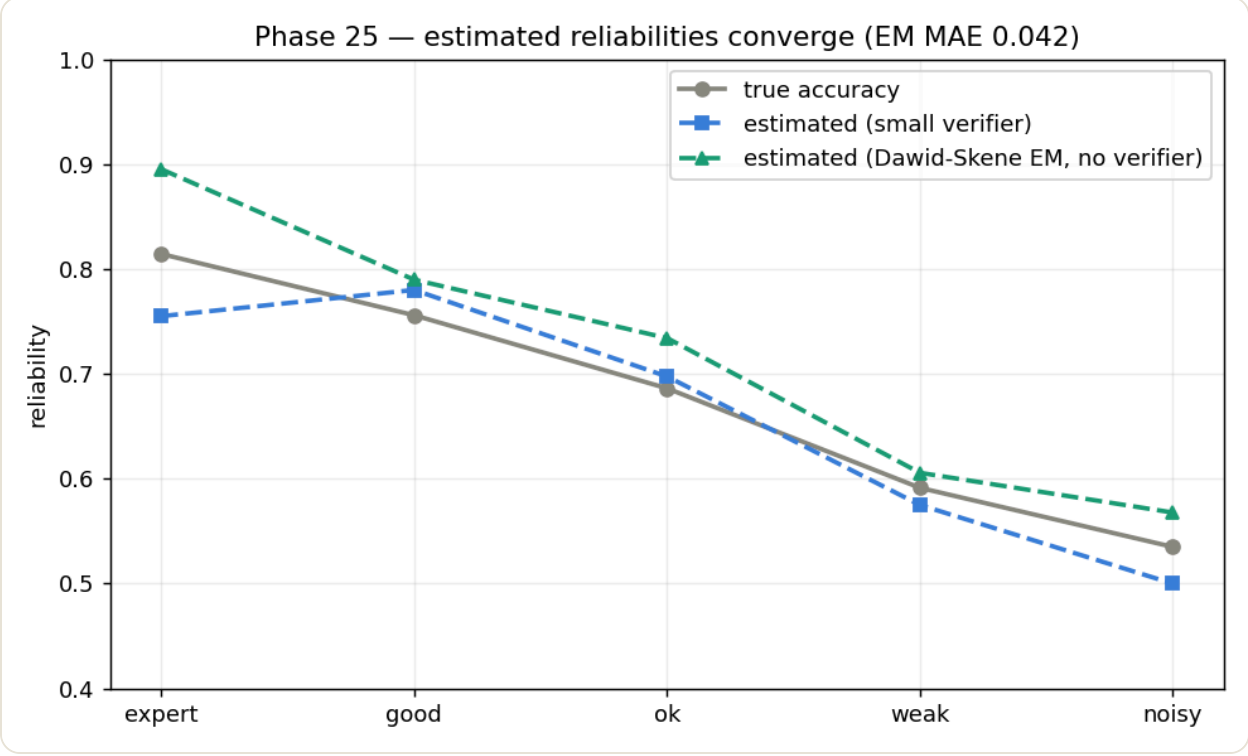


Figure 27. Per-teacher reliability is recovered from agreement patterns alone, with no answer key.

What the experiment showed

Reliability was recovered to within **0.042** with no ground truth; reliability-weighted consensus **inverted an adversarial panel where the unreliable outnumbered the reliable** (0.652 → 0.778). Drift was flagged at **latency 33 versus 300** for a fixed window, with **zero false alarms**. The keystone: principle inference reached **0.814 versus the teacher's own 0.714** — *the student exceeded its imperfect teacher* — beat behaviour cloning (0.675), and transferred across a teacher swap **with zero re-learning** while cloning collapsed.

INTERACTS WITH Generalizes the "learn from the slow model" signal that all earlier faculties assumed; its robustness is what makes the live-teacher stage (Arc X) survivable.

LIMITS & FUTURE Teachers here are a structured synthetic panel; a panel of genuinely diverse real LLMs is the natural next stressor.

CORE FACULTY 8 · ARC VIII · PHASES 28–32

F8 Open-Ended Reasoning (no verifier)

Reason well when answers *cannot* be checked — where quality comes from a latent judge, future human evaluations, or long-term outcomes rather than an answer key.

THE PROBLEM F3 worked precisely *because* its answers were checkable. Most real reasoning (Is this essay good? Is this plan wise? Is this design tasteful?) has no exact checker.

WHY OBVIOUS FIXES FAIL Without a verifier you cannot tell a good answer from a confident bad one. Training on a cheap *proxy* metric optimizes the proxy and lands credit on spurious features.

Biological seed

Prefrontal deliberation; default-mode self-reflection; deliberative System 2.

MECHANISM Five tools, each replacing a missing verifier with a *learned* one: **self-critique** (a learned critic predicts your own errors; revise only the flawed — cf. Self-Refine and Reflexion (**Madaan et al., 2023**; **Shinn et al., 2023**)); **internal debate** (specialists argue; synthesize by confidence × reliability — cf. AI-safety-via-debate (**Irving et al., 2018**)); **outcome-based judgment** (train on the true downstream result, not a proxy); **taste** (predict *future human evaluation*, as in learning from human preferences (**Christiano et al., 2017**)); **ambiguity** (answer as a calibrated set, separating genuine ambiguity from ignorance).

WORKED EXAMPLE

Asked to write, Cere drafts, *critiques its own draft* the way an editor would, and revises — but only where the critic flags a real problem, so it does not "fix" what was already good.

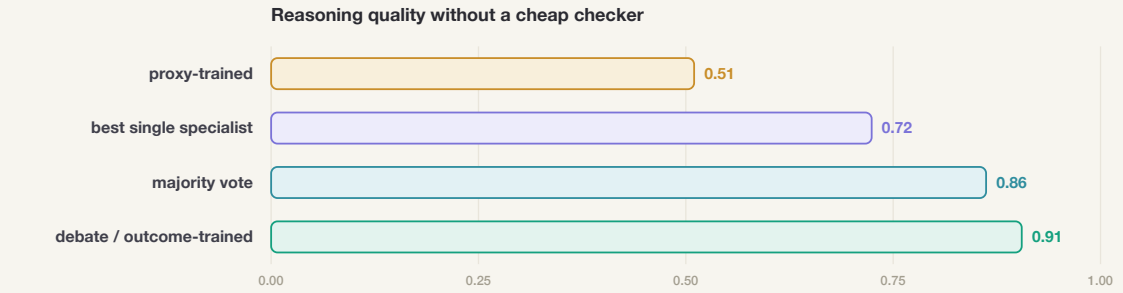


Figure 28. When answers cannot be checked, learned verifiers — debate, outcome training — recover quality a proxy metric cannot.

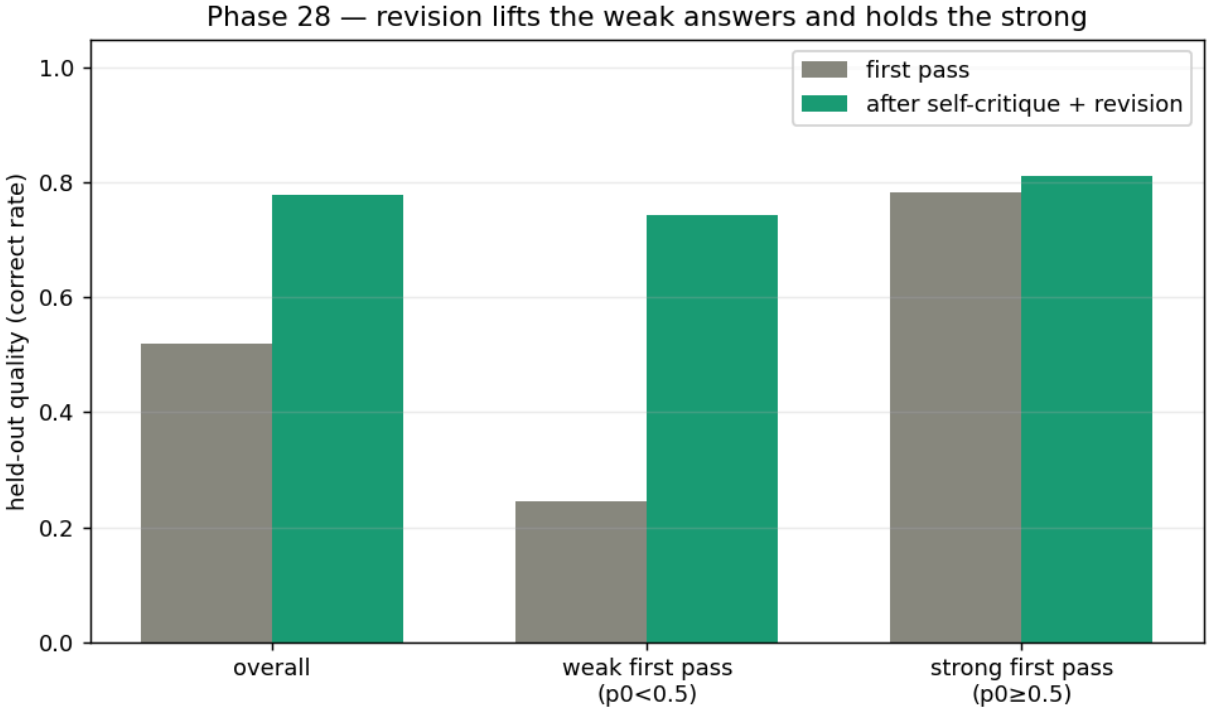


Figure 29. Gated self-critique: revise only what a learned critic flags. Quality rises and good answers are kept — naive regeneration degrades them.

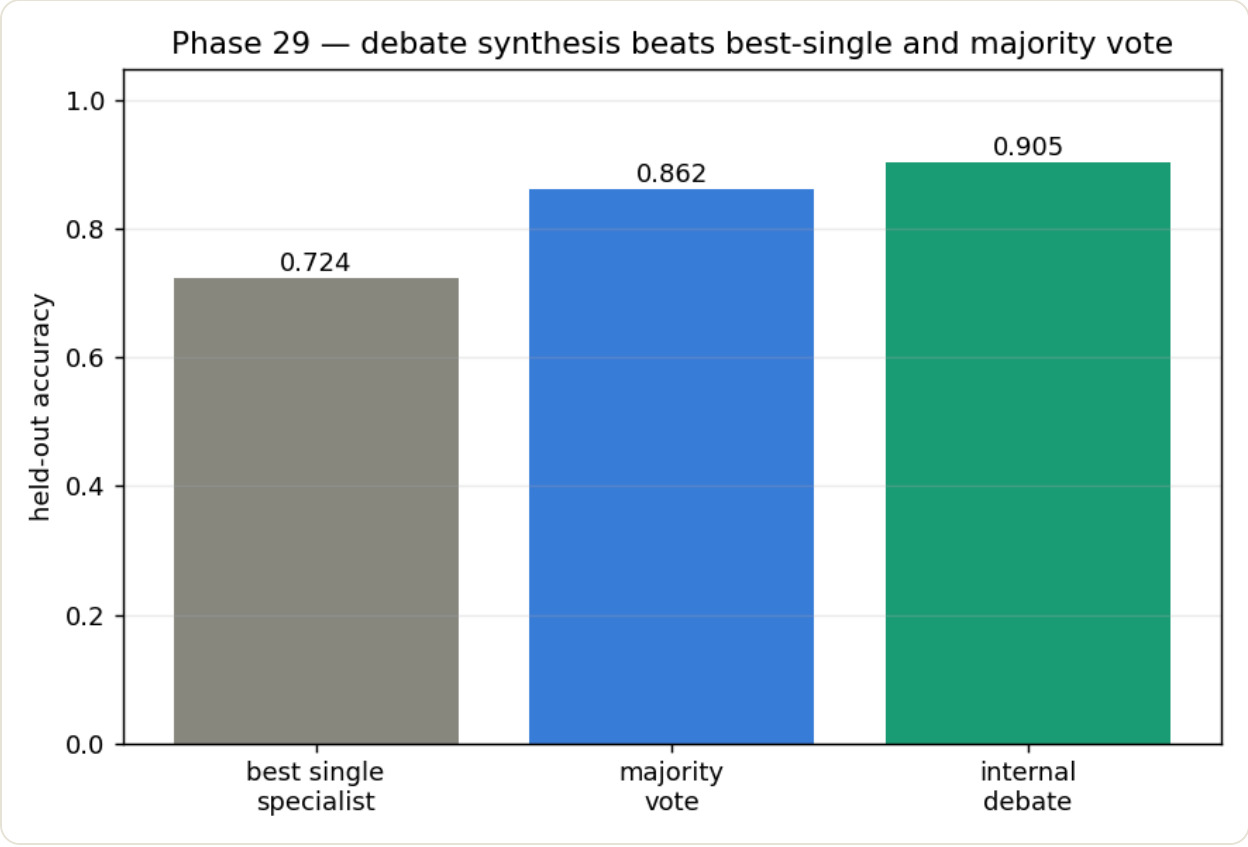


Figure 30. Internal debate among diverse specialists beats both majority vote and the best single specialist; the gain scales with diversity.

What the experiment showed

Gated self-critique lifted held-out quality **0.521 → 0.779** while retaining 100% of good answers (naive regenerate-everything degraded them to 0.69), the critic beating the generator's own confidence (AUROC 0.823 vs 0.726). Debate scored **0.905 versus 0.862** for majority vote and 0.724 for the best single specialist. Outcome-trained reasoning beat proxy-trained on the *true* metric (**0.876 vs 0.510**). A taste model predicted *future* human preferences at **0.918**. Ambiguity-as-a-set hit 0.91 coverage and separated ambiguity from ignorance perfectly (AUROC 1.00).

INTERACTS WITH Extends F3 to the uncheckable; its weak-verifier gating is what makes the no-verifier wild-stage (Arc X) hold quality.

LIMITS & FUTURE The "human evaluations" are a modelled annotator panel; real human-preference data is the next step.

F9 Executive Control & the Safety Core

Tie every faculty together under one executive that escalates to the big model *only* when it expects to be wrong or faces an untrusted irreversible step — inside an immutable safety boundary that no other faculty, not even the self-improving ones, may modify.

WHY IT MATTERS

A pile of clever modules is brittle; a single sealed loop is the opposite. F9 is where the project shows the parts can act as a *mind* — and stay safe under composition.

THE PROBLEM

A pile of faculties is not a mind. Something must decide what to do next, arbitrate competing pulls, and guarantee the safety invariants hold *simultaneously*, under composition — not just one at a time.

WHY OBVIOUS FIXES FAIL

If safety is a property each faculty promises individually, composition can still break it (faculty A's "safe" action is unsafe given faculty B's state). Safety has to be a *separate, sealed* mechanism the others operate inside — the same logic as shielding in safe RL (**Alshiekh et al., 2018**), and a direct response to the avoid-negative-side-effects / safe-exploration problems catalogued for AI safety (**Amodei et al., 2016**).

Biological seed

Prefrontal central executive and goal arbitration; inhibitory control — a deliberately loose analogy; the property that matters is that it is *sealed*, not that it is biological.

MECHANISM

A **self-generated-goal** selector allocates practice to whatever is improving fastest. The **Executive** composes cache+policy, value, wrongness, and the reversibility gate into one online runtime, demoting the big model to court of appeals. The **Safety Core** — the irreversible-action effect gate plus the certification registry — is armed from boot, reused exactly, and is *non-evolvable*.

WORKED EXAMPLE

Over a long, varied workload, the executive quietly handles the routine, appeals the genuinely novel, and — when you break the fast path on purpose — the failure shows up as *more appeals (cost)*, never as a wrong or unsafe action.

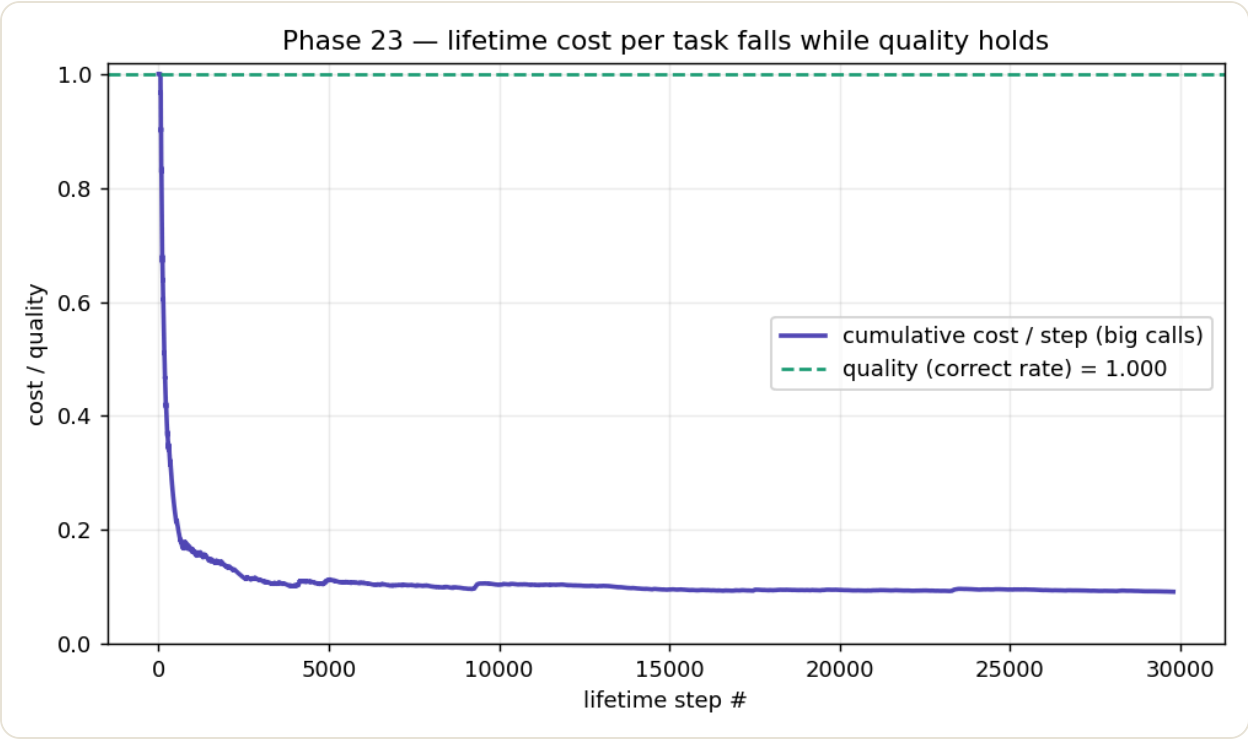


Figure 31. The Executive capstone over a 9,000-step lifetime: escalation to the big model falls to the genuine-novelty floor while quality holds at 1.000.

Phase 23 — full regression suite holds under the unified Executive

regression invariant	verdict	detail
calibration (value ECE<0.12)	PASS	ECE 0.027
selective prediction (wrong>=margin)	PASS	AUROC 0.990 vs 0.988
no-forgetting (Δacc>=-0.02)	PASS	min Δ +0.000
safety gate (0 unsafe commits)	PASS	unsafe 0
failure isolation (quality holds)	PASS	q 1.000, esc 0.26, unsafe 0
no-retrain (online + dream only)	PASS	by construction

Figure 32. The full Arc I–IX regression suite, held simultaneously — composition is the hardest thing to prove, and it holds.

What the experiment showed

Over a 9,000-step open-ended lifetime, **escalation fell 0.57 → 0.07** (the genuine-novelty floor) while quality held at **1.000** and cost per step fell 0.112 → 0.092. The **entire Arc I–IX regression suite held at once**: calibration, selective prediction, no-forgetting, *both* safety gates with **0 unsafe commits**, failure isolation, and no retraining from scratch. On the real-teacher version (Phase 38) the same held, byte-for-byte reproducibly, with 100% of late consults being genuine cache misses.

INTERACTS WITH *Everything* — it is the integration point. The safety core is referenced by every action-taking faculty and sealed against every self-modifying one.

LIMITS & FUTURE Proven on minimal, synthetic substrates and a cached real teacher; a live, large-scale executive is the central integration target.

THE HIGHER-ORDER FACULTIES · GROUPED

Beyond the nine core organs

The roadmap turned the *same* consolidation–value–gate machinery on progressively higher-order cognition. These are built and GO as isolated mechanisms; they are summarized here because each is a variation on machinery already explained.

Knowledge creation · Arcs XII–XXI

Can the system form *new* knowledge, not just reproduce a teacher's? **Hypothesis formation** runs a full observe → hypothesize → experiment → update loop, recovering hidden rules far better than random querying. **Abstraction** and **meta-abstraction** compress recurring structure into named concepts, then concepts into a few organizing principles. **Creativity** mutates and recombines ideas under a calibrated critic. **Scientific discovery** and **discovery engines** compress a literature, find its gaps, and plan research programs. **Invention**, **self-improvement**, and **cognitive evolution** let the system invent, test, and adopt new mechanisms — *behind the sealed safety core*: self-improvement compounded competence (0.759 → 0.831) while the core held every round.

Social cognition · Arcs XX, XXV, XXVIII

Theory of mind infers other agents' beliefs, goals, and intentions — including *false* beliefs — turning that inference into cooperative gain while isolating a deceptive agent. **Social interaction** learns to signal, negotiate, cooperate under mixed motives, and build reputation-based trust. **Collective specialists** let specialists *within one Cere* form, exchange knowledge with zero private leakage, and reason as a team beyond any single member.

Understanding & judgment · Arcs XXIV, XXVI, XXVII

Causal understanding moves from "what follows what" to "what *causes* what," supporting intervention and counterfactual reasoning that survives a surface shift where correlation collapses. **Common sense** learns transferable everyday physical / social / temporal knowledge that cuts slow calls. **Wisdom** balances long-term, irreversible, multi-stakeholder tradeoffs and — pointedly — learns *restraint*, knowing when **not** to act; throughout, judgment *sharpens* the mechanical gate but never *replaces* it.

> **The shared contract**
>
> Every grouped faculty shares the core's non-negotiables: it is learned online, it is calibrated, its irreversible steps stay behind the sealed gate, and it is validated on a minimal substrate against named baselines.

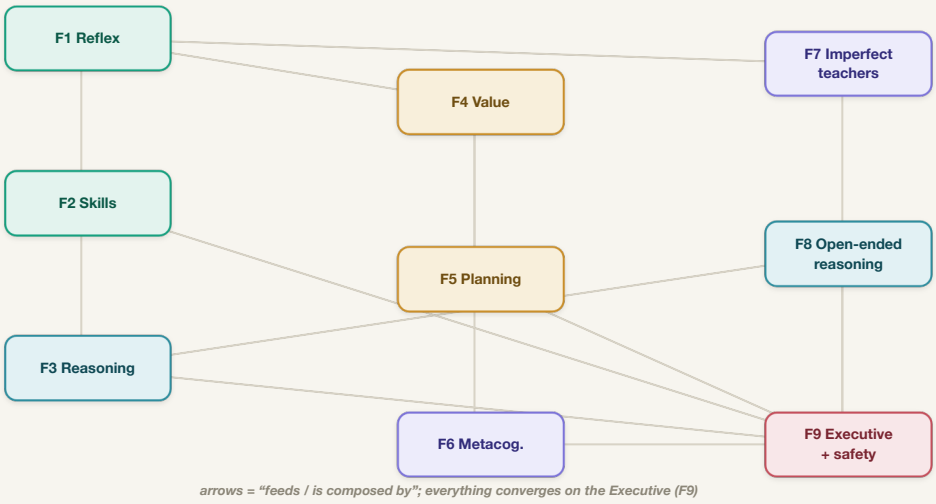


Figure 33. The faculty interaction graph. Reflex (F1) underlies skills (F2) and reasoning (F3); value (F4) feeds planning (F5) and metacognition (F6); robust learning (F7) and open-ended reasoning (F8) generalize the teacher signal — and everything converges on the Executive (F9), which holds the sealed safety core.

KEY TAKEAWAYS

- Nine core faculties, each the same loop pointed at new content: reflex, skills, reasoning, value, planning, metacognition, robust learning, open-ended judgment, and the executive.
- Faculty 3 is the pivot — the same machinery that made actions reflexive makes reasoning reflexive (speculative cognition).
- Robust distillation lets the student *surpass* an imperfect teacher and survive its replacement; open-ended reasoning works without a cheap checker.
- Higher-order faculties (knowledge creation, social cognition, judgment) reuse the identical machinery and keep the same safety contract.

PART IV

Two hundred experiments are not 200 results — they are 39 questions

The Experimental Story

Each arc is a deliberate step that drops one comforting assumption the previous step relied on. This part tells that story in order: what question was on the table, and what the experiment settled.

WHAT YOU WILL LEARN

- The 200-phase project as 39 research questions, each dropping one assumption.
- The single most important sanity check: the thesis confirmed on real models.
- The master map — every phase to its arc, labelled faculty / stage / system.
- How the eras build: from "can a fast model learn a slow one" to "can it discover its own missing faculties."

WHY THIS MATTERS

Part III asked "what is each organ?" Part IV asks "what question was on the table, and what did the experiment settle?" The arcs are ordered so that each removes a comfort the last one leaned on — a correct teacher, a checkable answer, a short horizon — until what remains looks like judgment.

KEY IDEAS

- Isolation-first: every mechanism tested alone, on the smallest substrate, against named baselines.
- A phase is "done" only when its named numbers hold — never merely because it runs.
- The real-model track shows the thesis is not a toy artifact.
- Composition is proven at capstones; integration at scale is the open work.

§ The method, in one principle

Isolation first. Every mechanism is built and tested *alone*, on the smallest substrate that can express it, against named baselines, with an explicit "done-when" criterion — *before* any integration. A phase is "done" only when its named figures or numbers hold, never merely because the code runs. The cost of this rigor is the project's central honest caveat (Part VII): the pieces are proven *individually*; assembling them into one large running system is the open work.

Concretely, every phase follows the same experimental template, and the faculty entries in Part III and the arc entries below are that template in compact form: a **research question** (the arc's title question) → a **hypothesis** (the mechanism will beat the baseline on a named metric) → a **baseline** (slow-only, linear extrapolation, greedy, blind search, behaviour cloning, majority vote, a fixed threshold, …) → a **method** (the mechanism) → **metrics** (success, slow-consult rate, ECE, AUROC; defined in Appendix B.3) → a **pass criterion** (the pre-registered "done-when") → a **result** (the figure / number) → a **conclusion** (proved / did-not). Read each faculty's *Problem* · *Mechanism* · *Worked example* · *Evidence* · *Limits* blocks as exactly these fields.

Figure 34. The research at a glance: six eras, 40 arcs, 205 phases. Eras 1–5 (200 phases) are built and GO in isolation; Era 6 is proposed. Each era either adds faculties, exercises them in a harder setting, or builds the tissue that integrates them.

IV.1 Era 1 — Learn from teachers (Arcs I–X)

This is the proven core, the part validated most rigorously and the only part that touches real models.

Arc I — Reflexive Prediction (P1–4)

> **§ The question on the table**
> Can a tiny model hide a slow model's latency by predicting the next action — and can the slow model then step back from executor to supervisor?

The trained predictor reached 74% where baselines were at 1–3%; slow-call rate fell to the irreversibility floor at unchanged success; the dream pass defeated forgetting; and a graduated supervisor's big-model call rate fell 0.68 → 0.26 with a **100% catastrophic-error catch rate**.

PROVED	Prediction hides latency; routine work migrates slow → fast; the safety floor is real; competence survives across sessions.
DID NOT	The environment is deterministic-given-context (so the cache is perfectly calibrated) and the teacher is a scripted oracle — calibration under a noisy teacher is left as the top open stressor.

The headline real-model test

Before going further, the project replaced the NumPy stand-ins with real models: a **Qwen2.5-0.5B** model as the fast path (with a lightweight LoRA adapter (Hu et al., 2021), on a laptop GPU) learning from **Claude-Haiku-4.5**

as the teacher — knowledge distillation (**Hinton et al., 2015**) from a frontier teacher into a tiny student — with the actions deliberately encoded as opaque codes so nothing was guessable.

0.09 → 0.91
top-1 action accuracy: chance → after training

≈ teacher
0.91 was the teacher's own ceiling

~4 epochs
to the ceiling, on a laptop

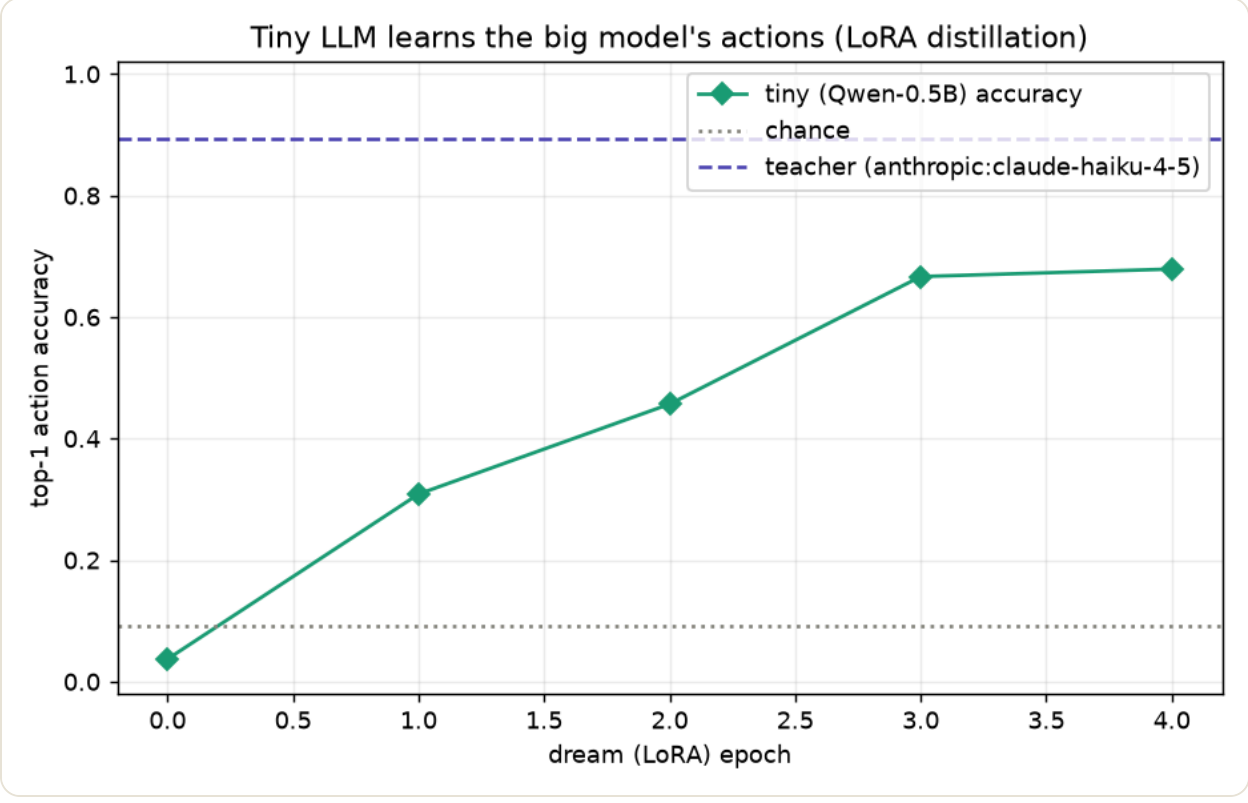


Figure 35. A 0.5-billion-parameter model learns a frontier model's behaviour from its labels alone, climbing from chance toward the teacher's ceiling over a few dream/LoRA epochs. This is the single most important sanity check in the project: the thesis is not an artifact of toy models.

Why this matters

A 0.5B model learned a frontier model's behaviour from its labels alone, to the teacher's competence, in about four training epochs on a laptop. Everything else in the report is built on minimal substrates *by choice* (for reproducibility) — but this test shows the core mechanism survives contact with real models.

Arcs II–III — Skill & Reasoning Consolidation (P5–13)

§ The question on the table

Can repeated *chains* of actions, and then repeated chains of *reasoning*, be compressed into one-shot skills?

Decisions per task fell 45% (actions) and reasoning steps fell 62% (thought), at held quality, with the *identical transfer-library code* working for both. **Arc III, Phase 9, is the pivot** where the project's claim widened from "learns actions" to "learns actions *and reasoning*" — speculative cognition.

PROVED Consolidation is *alphabet-agnostic*; the safety gate cleanly flips from reversibility (actions) to verification (thoughts); the mechanisms *compose* over a mixed lifetime without forgetting.

DID NOT The two stacks remain separate libraries (a single merged library and one dream spanning both is later work); substrates are minimal and synthetic.

Arcs IV–VI — Judgment, planning, and the executive (P14–23)

§ The question on the table

Beyond reproducing the teacher's actions, can Cere learn the teacher's *judgment* — value, search, calibration — and pull every faculty under one executive?

A calibrated value head beats policy confidence at gating; value-guided imagination crushes greedy and blind search; meta-cognition predicts its own wrongness far better than a hand-tuned margin; judgment *transfers* across domains; and the **Executive capstone (P23)** runs the whole stack with escalation falling to 0.07 at quality 1.000 while the full regression suite holds at once.

PROVED Judgment is a separable, learnable layer; **the big model becomes a court of appeals** — the originally-stated destination, reached and extended.

DID NOT Each win needs a non-stationary probe to surface; a stationary stream hides the value-add. Systems-scale unification is future work.

Arcs VII–IX — Imperfect teachers, no verifier, deep planning (P24–37)

§ The question on the table

What happens when we drop the last comforts — a *correct* teacher (VII), a *checkable* answer (VIII), and a *short* horizon (IX)?

The student learned to surpass an imperfect teacher and survive its replacement; to reason well with only weak verifiers (self-critique, debate, outcome, taste, calibrated ambiguity); and to plan far ahead via hierarchy, abstraction, pruning, counterfactuals, and long-horizon credit. With Phase 37, **all 37 core-faculty phases across nine arcs were built and GO.**

IV.2 The capstones — does it all compose?

The grand capstone (P38)

> **§ The question on the table**
>
> Do the faculties actually compose, on real teacher behaviour, over a lifetime?

Phase 38 folded every faculty from all nine arcs into one runtime around a **cached real teacher** and a live tiny fast path, over a 9,000-task non-stationary lifetime. Consult rate fell 0.57 → 0.07 onto the novelty floor, every Arc I–IX invariant held *simultaneously*, and two runs hashed identically. It is evidence for **composition + reproducibility** on this substrate, not for new scale.

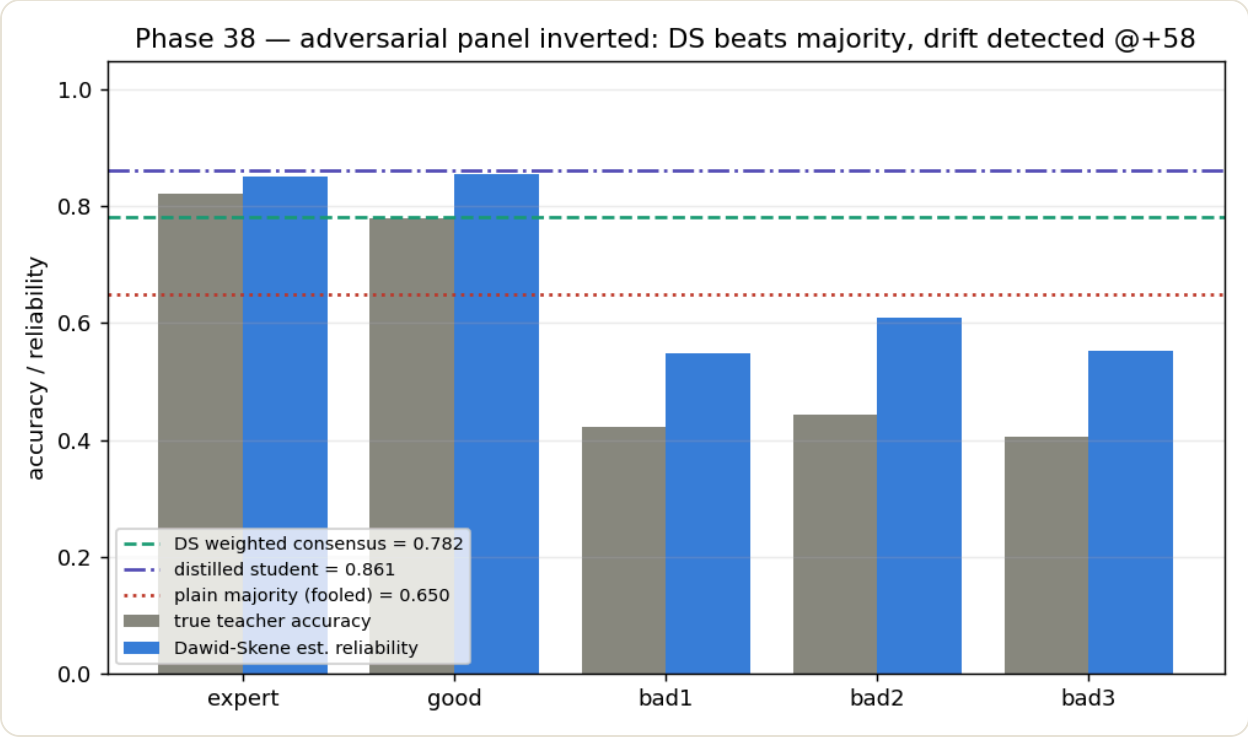


Figure 36. The grand capstone (P38): every faculty in one runtime over a 9,000-task lifetime against a cached real teacher. Consults fall to the novelty floor; the full regression suite holds at once; two runs hash identically.

Arc X — Court of Appeals in the wild (P39–50)

> **§ The question on the table**
>
> Off the minimal substrate: in the LLM tool-agent setting (**Yao et al., 2022**; **Schick et al., 2023**; **Yang et al., 2024**), can the big model stay mostly out of the loop on real-style tasks, with real irreversible actions, a *live drifting* teacher, multi-user isolation, and real models?

The capstone (P48) ran a ~12,000-step multi-user, drifting, open-world lifetime: **quality 0.994 (≥ teacher), 0 unsafe irreversible commits, slow-consult 0.193 → 0.097**, appeals 100% explained, drift detected and recovered, no forgetting. Live **Haiku-4.5 agreed 0.981** with the oracle as the appeals court; real **Qwen-0.5B distilled Haiku 0.09 → 0.68**; cross-user leakage was **0**.

PROVED

The big model is no longer the default executor — it is the appeals court, and that holds with real tools, real irreversible actions, real drift, real isolation, and real models, at a reproducible scale.

DID NOT

The long committed runs use the free oracle as the reproducible teacher (real Haiku wired via `--api`, proven at 0.981 agreement) — this is "survives the wild at reproducible scale," not "production deployment."

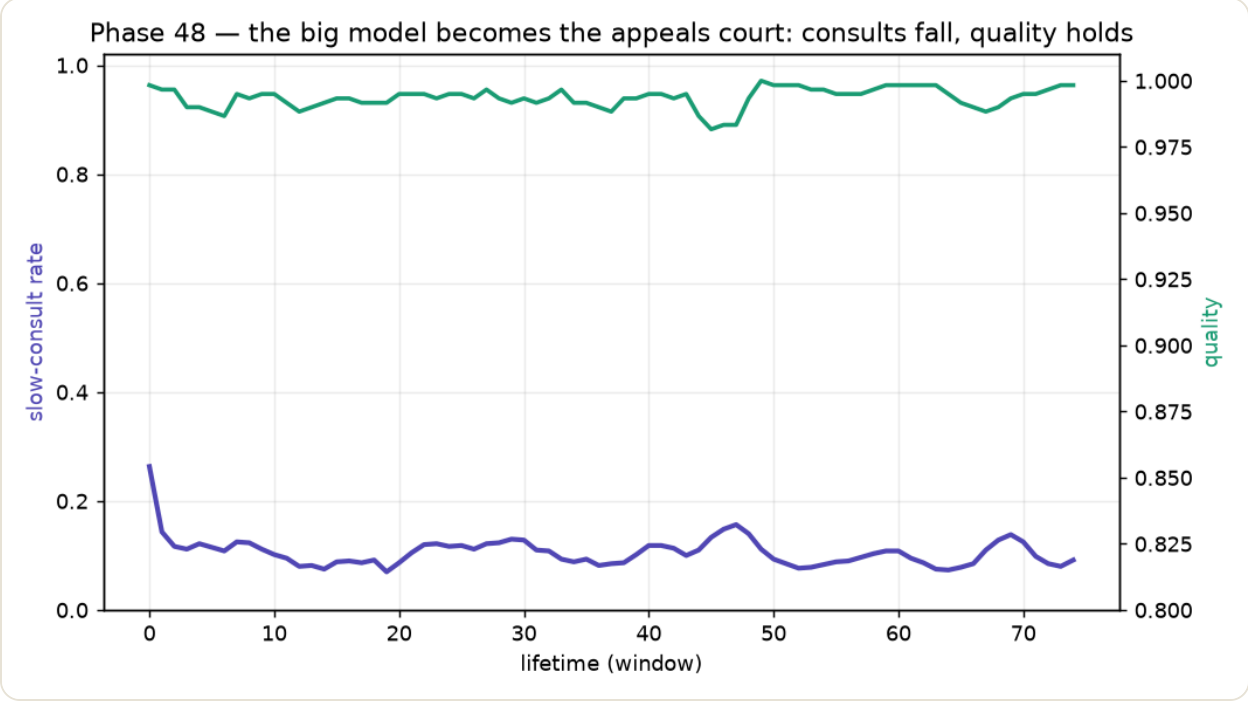


Figure 37. Arc X capstone (P48): a multi-user, drifting, open-world lifetime on six real-style tool families. Slow consults fall by half onto the novelty floor while quality stays at or above the teacher and unsafe commits stay at zero.

IV.3 Eras 2–5 — beyond the wild

Past the Court of Appeals, the roadmap turned the same machinery on higher-order cognition and harder settings. All were built and GO as isolated mechanisms. A representative pass:

Movement	Arcs	The question each answered (and the headline)
Deployment bridge	XI (P51–55)	*Can the split do real office work?* Browser/shell/office work migrates slow→fast (0.44→0.18) at 0 unsafe; long-running memory drives slow-calls toward 0 behind a 0-leakage firewall.
Forming new knowledge	XII–XVI	*Hypotheses, abstractions, creativity, discoveries?* Closed loops recover hidden rules (0.99 vs 0.80 random); directed creativity beats undirected; self-improvement compounds behind a sealed core.
New machinery	XVII–XXI	*Can it invent and evolve its own mechanisms?* Invented/evolved mechanisms beat their parents, all valid, runnable, and safe-core-preserving (auto-bench predicts true performance at r≈0.997).
Embodiment (sim)	XXII	*Can it learn the physical world?* Forward prediction under sensor noise, object permanence, motor primitives, collision-free navigation — behind a fail-closed physical gate, sim-first.
Autonomy (stage)	XXIII	*Can it choose its own work, safely?* A goal compiler + an Executive Board of competing proposers under a hard Safety veto; competence compounds 0.30→0.95, 0 out-of-contract.
Deeper faculties	XXIV–XXVIII	*Cause, other minds, common sense, wisdom, society.* Each beats its associative / asocial / myopic baseline; judgment sharpens but never replaces the gate.
Hardest settings	XXIX–XXX	*Can many Cere cooperate, and one improve for a lifetime?* A leak-free collective solves beyond any member; the Phase 155 grand capstone holds a full Arc I–XXX regression suite — 12/12 invariants — at once.
Integration & self-extension	XXXI–XXXIX	*Can the faculties become one mind, and can it discover its own missing ones?* Era 4 builds the coordination systems; Era 5 rediscovers 5/5 withheld faculties, human-gated, with 0 unauthorized installs (see Part VIII).

Table 2. Eras 2–5 in one table. The through-line: no later arc invents a new safety story — self-modification and self-direction all hold the same sealed, non-evolvable core.

IV.4 The master map — every phase to its arc

The table below maps all 200 phases to their 39 arcs, labelled by class (**F**aculty / **S**tage / **Sys**tem) and status. Every phase falls in exactly one arc; the capstone phase of each arc is the one whose “done-when” is the arc's headline.

Era	Arc	Phases	Class	Faculty / setting
1	I — Reflexive Prediction	1–4	F	Forward prediction & reflex
1	II — Skill Consolidation	5–8	F	Compress actions into skills
1	III — Reasoning Consolidation	9–13	F	Compress reasoning (the pivot)
1	IV — Intuition & Planning	14–16	F	Value, guided search, calibration
1	V — Meta-Cognition	17–19	F	Wrongness, budget, strategy
1	VI — Open-Ended Intelligence	20–23	F	Transfer, unified latent, **Executive**
1	VII — Imperfect Teachers	24–27	F	Disagreement, reliability, drift, principle
1	VIII — Open-Ended Reasoning	28–32	F	Critique, debate, outcome, taste, ambiguity
1	IX — Deep Planning	33–37	F	Hierarchy, abstraction, prune, credit
1	(grand capstone)	38	—	Every faculty, cached real teacher
1	X — Court of Appeals	39–50	S	Real tools, irreversible actions, live models
2	XI — Knowledge Work	51–55	S	Browser, shell, docs, cross-tool memory
2	XII — Hypothesis Formation	56–60	F	Observe→hypothesize→experiment→update
2	XIII — Abstraction Generation	61–65	F	Concepts, hierarchy, distillation
2	XIV — Creativity	66–70	F	Mutation, recombination, directed
2	XV — Self-Improvement	71–75	F	Mine→propose→sandbox→adopt (sealed core)
2	XVI — Scientific Discovery	76–80	F	Literature, gaps, novel hypotheses
2	XVII — Invention	81–85	F	Mine/mutate/recombine mechanisms
2	XVIII — Meta-Abstraction	86–90	F	Principles across domains
2	XIX — Discovery Engines	91–95	F	Frontier, opportunity, portfolio
2	XX — Collective Specialists	96–100	F	Specialists within one Cere, 0 leakage
2	XXI — Cognitive Evolution	101–105	F	Evolvable genome (sealed core)
2	XXII — Embodied Intelligence	106–110	S	Physical world, sim-first
2	XXIII — Initiative & Autonomy	111–120	S	Goal compiler, Executive Board
2	XXIV — Causal Understanding	121–125	F	Discovery, intervention, counterfactual
2	XXV — Theory of Mind	126–130	F	Belief/goal inference, false belief
2	XXVI — Common Sense	131–135	F	Physical/social/temporal plausibility
2	XXVII — Wisdom	136–140	F	Horizon, irreversibility, restraint
2	XXVIII — Social Intelligence	141–145	F	Signalling, negotiation, trust
3	XXIX — Collective Intelligence	146–150	S	Many full Cere cooperate, 0 leakage
3	XXX — Long-term Open-Ended	151–155	S	Multi-year lifetime; **grand capstone**

Era	Arc	Phases	Class	Faculty / setting
4	XXXI — Global Workspace	156–160	Sys	Broadcast bus + ignition (keystone)
4	XXXII — Attention Routing	161–165	Sys	Bandwidth-bounded gating
4	XXXIII — Episodic Memory	166–170	Sys	One-shot binding, index, replay
4	XXXIV — Semantic Memory	171–175	Sys	Episodic→semantic, queryable
4	XXXV — Neuromodulation	176–180	Sys	Four bounded control axes
4	XXXVI — Salience & Homeostasis	181–185	Sys	Interoception, engage/idle
4	XXXVII — Identity & Self-Model	186–190	Sys	Persistent, corrigible “I”
5	XXXVIII — Cognitive Development	191–195	F	Detect *which faculty is missing*
5	XXXIX — Open Cognitive Evolution	196–200	F	Build & install it, human-gated

Table 3. All 200 phases across 39 arcs. Every one is built and GO on a minimal, reproducible substrate in isolation (through 2026-06-26). Class: F = faculty, S = stage, Sys = system.

KEY TAKEAWAYS

- 200 experiments = 39 questions, each dropping one assumption the last relied on.
- The real-model test (Qwen-0.5B ← Haiku-4.5, 0.09 → 0.91) shows the thesis survives real models.
- Capstones (P23/38/48/155/200) prove the faculties *compose* — many at once, reproducibly.
- Every phase is GO in isolation; folding them into one live system is the open work.

PART V

There is exactly one loop, and it never changed

The Learning Loop

Two hundred experiments, thirty-nine arcs, five eras — and every one of them is the same seven-step cycle, with a different organ plugged into one of its slots. This is the project's deepest structural result.

> *A pile of clever modules is brittle; every new module is a new way for the others to break. A single loop with a sealed safety step is the opposite: every new module is just new content in an old, well-tested slot.*

WHAT YOU WILL LEARN

- The single seven-step loop, stated as a contract every faculty must honour.
- How each of the nine faculties is the *same* loop with different content in one slot.
- The three reasons the loop stays valid as the system scales to 200 phases.
- The composition test — proving all the invariants hold at once, not one at a time.

WHY THIS MATTERS

If you remember one thing about Cere, remember this: the loop is the constant. Understanding why a 200-phase architecture could keep its foundation is the same as understanding why the loop never had to change.

KEY IDEAS

- Perceive → Predict → Verify → Act → Outcome → Learn → Consolidate.
- A faculty is "well-formed" only if it fits the loop without breaking an invariant.
- The safety floor is a *separate, sealed* step the faculties pass through.
- Graceful degradation: the worst case is slowness, not catastrophe.

1 The loop, as a contract

Every faculty is the same seven-step cycle from Part II, with a different organ plugged into one of its slots. Read the steps as a contract that every faculty must honour.

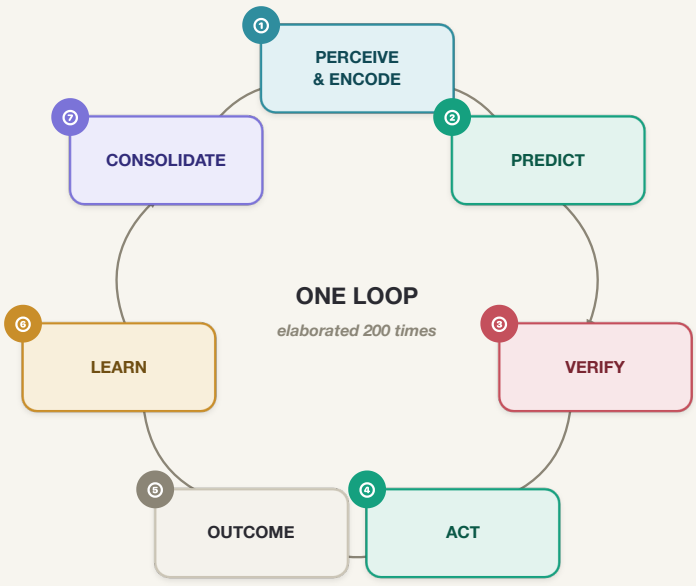


Figure 38. The one loop. Perceive and encode the situation; the fast path predicts; the gate verifies; act or appeal; observe the outcome; learn from the error; consolidate so nothing is forgotten — and back to the start, now faster here.

Step	The question it answers	The invariant it must preserve
① Perceive/Encode	“What situation am I in?”	downstream depends only on the *encoded context*, never raw input
② Predict	“What's the fast answer?”	the forward pass stays far under the latency budget
③ Verify	“Can I trust the fast answer?”	irreversible-not-trusted ⇒ appeal (the safety floor)
④ Act	“Do it (or appeal).”	reversible / simulated / read-only only, unless certified
⑤ Outcome	“What happened?”	the real result is observed, not assumed
⑥ Learn	“Train on the gap.”	online + dream only — *never* a from-scratch retrain
⑦ Consolidate	“Don't forget the old.”	no catastrophic forgetting; the cache never forgets

Table 4. The loop as a contract. A new faculty is “well-formed” only if it can be expressed as a new slot-filler without violating any invariant — the actual acceptance test the project applied.

2 Every faculty is the same loop, re-pointed

Each faculty changes *what flows through a slot*, not the slot structure.

Faculty	What it changes in the loop
Forward prediction	② predicts the next **action**
Skill consolidation	② predicts a whole **chain** of actions in one shot
Reasoning consolidation	② predicts a **reasoning** chain; ③ verifies the **answer**
Intuitive valuation	adds “how good / risky?” to ② and ③
Guided imagination	② becomes imagine→score→explore over a world model
Metacognition	③ gates on a **learned** wrongness, not a hand threshold
Imperfect teachers	⑥ learns from a **noisy** a*, inferring the principle
Open-ended reasoning	③ uses **weak verifiers** when no checker exists
Executive + safety	③/④ composed under one runtime; the gate is **sealed**
Causal / ToM / wisdom	richer features into ②; sharper judgment into ③

Table 5. In every row, steps ①, ⑤, ⑥, ⑦ are unchanged. The loop is the constant; the faculty is the variable.

The interception predictor and the scientific-discovery engine are the same seven steps with different content in steps ② and ③.

3 Why the loop stays valid

Reason 1 — the slots are content-agnostic

Everything downstream of encoding depends only on an abstract *context → action/answer* mapping, so the machinery genuinely does not care whether the tokens are ball positions, tool calls, reasoning ops, or hypotheses. The most pointed evidence is Phase 10, where the *same transfer-library code*, byte-for-byte unchanged, consolidated reasoning fragments exactly as it had consolidated tool fragments. When the same code works on two alphabets without edits, "alphabet-agnostic" stops being a claim and becomes an observation.

Reason 2 — the safety floor is a separate, sealed step

Verification (③) is not woven into the faculties; it is its own mechanical step that they all pass through. That separation is what lets the *same* gate protect a brand-new faculty — and what lets the self-improving arcs be told, structurally, "you may change ②, never ③." A safety property that lives in one sealed place is a property you can actually keep.

Reason 3 — learning is always online + dream, never retrain

No faculty is allowed to "start over." New competence is added by online updates (**Cesa-Bianchi & Lugosi, 2006**) and protected by the dream pass — the lifelong / continual-learning regime (**Thrun, 1998**; **Parisi et al., 2019**), with a curriculum that grows with competence (**Bengio et al., 2009**). This is what makes competence *accumulate* rather than reset, and it is checked explicitly in every capstone.

4 The loop under composition

The single hardest thing to prove is not that each faculty works alone, but that they **all work at once without interfering.** A value head calibrated alone might be knocked out of calibration when a planning module feeds it strange states; a safety gate that holds for actions might be bypassed by a reasoning path. The capstones exist precisely to test this.

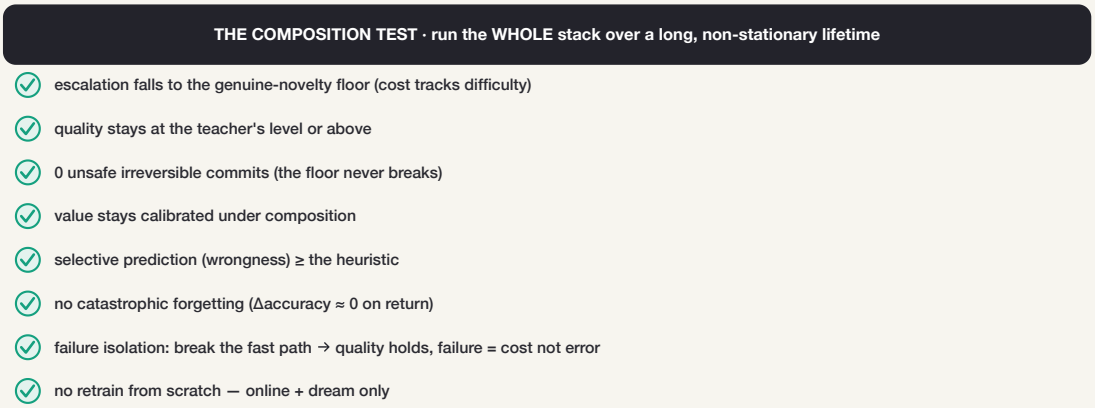


Figure 39. The composition test, run at Phases 23, 38, 48, 155, and 200. All eight invariants must hold simultaneously over a long, non-stationary lifetime — and at every scale the project could build, the suite holds.

The deepest property: graceful degradation

Break the fast path on purpose and, in these experiments, escalation rises while quality stays at 1.000 — the failure converts into *more appeals*, not into a wrong or unsafe action. **The worst case we observed is slowness, not catastrophe.** That is the reason the architecture can keep absorbing faculties without becoming fragile.

Cere bet that the loop, not any individual faculty, is the durable thing — and 200 phases of "the regression suite still holds" is the evidence that the bet paid off.

KEY TAKEAWAYS

- One seven-step loop underlies all 200 phases; faculties change a slot's content, never the structure.
- The loop is content-agnostic, its safety step is sealed and separate, and learning is always online + dream.
- Composition is the hard part — and the full regression suite holds at every capstone.
- Graceful degradation: a broken fast path becomes cost, not catastrophe.

PART VI

A proof is not a product — the other half is a living system

From Research to Runtime: CereOS

The 200 phases prove faculties, one at a time, in isolation. But a directory of validated mechanisms gives you nothing to use. CereOS is the Live Cognitive System that hosts proven faculties and keeps them alive as one persistent runtime you can talk to.

WHAT YOU WILL LEARN

- The two tracks the project runs on — research and runtime — and the one rule that keeps them honest.
- How a proven faculty becomes live cognition: registry → adapter → ThinkingLoop seam.
- The runtime lifecycle of a single turn: encode → think → execute → result.
- Where CereOS is today on its v0 → v7 capability axis, with honest caveats.

WHY THIS MATTERS

Research answers "can a small model learn this?" CereOS answers "how does proven cognition become something you use, persistently?" The discipline is that the runtime never invents cognition — it only hosts what a phase has already proven.

KEY IDEAS

- CereOS does not invent cognition; it hosts proven mechanisms.
- Capability grows by adding faculties to a registry — never by editing the spine.
- The safety core is armed from boot and reused exactly.
- The runtime boots only faculties that pass their eval — status is verified, not trusted.

◐ The difference between a discovery and an organism

The research arcs are like discovering, one by one, how a heart, a liver, and a pair of lungs work — each on its own bench, each proven to function. CereOS is the body that hosts them: a single nervous system where perception, routing, memory, executive coordination, cognition, action, and maintenance run *together*, continuously, so the whole thing stays coordinated and alive rather than firing once and halting.

1 Two tracks, one rule

The project deliberately runs on two tracks that must not be confused.

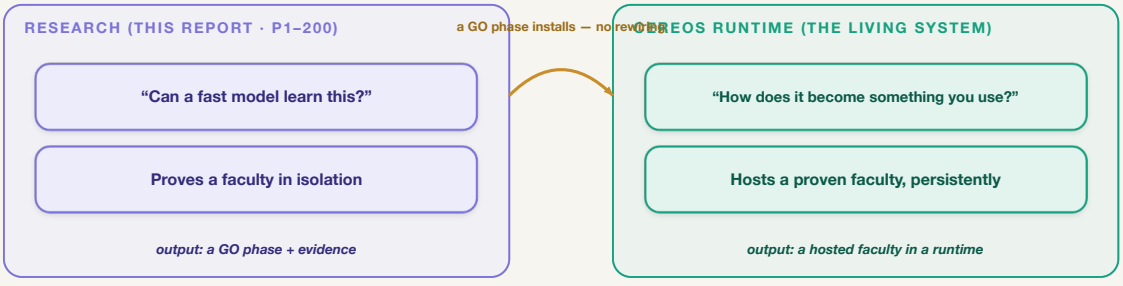


Figure 40. Two tracks. Research validates a faculty in isolation and emits a GO phase; CereOS installs that GO phase as a hosted faculty, with no rewiring of the executive spine.

§ The one rule

CereOS does not invent cognition. A new learnable mechanism is **research** — prove it in a phase first. CereOS builds only the runtime structure that *hosts* a proven mechanism and swaps a stub for it with no rewiring. The swap point is a single seam called the **ThinkingLoop**. Capability grows by *adding faculties to the registry*, never by editing the executive spine — and the safety core is armed from boot and reused exactly.

2 How a proven faculty becomes live cognition

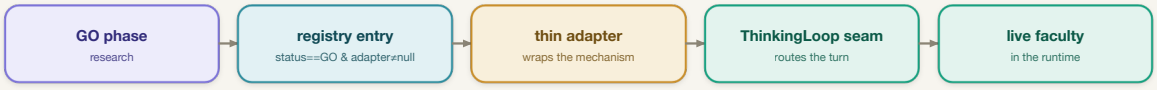


Figure 41. From research to live cognition. The runtime boots from a registry — one row per faculty, loaded only when it is genuinely GO and has a real adapter. The runtime runs each faculty's eval at boot and refuses any that is not green, so status honesty is verified, not trusted.

A turn flows **encode → think → execute → result**: the encoder turns a real request into a decision context, the ThinkingLoop decides fast-path-or-appeal, and an executor turns the selected action into the user-facing reply.

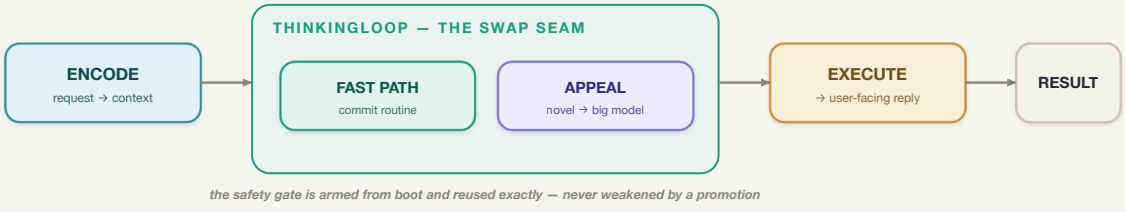


Figure 42. The runtime lifecycle of a single turn. The ThinkingLoop is the one seam where "how to decide" plugs in; the safety gate sits underneath it, armed from boot.

3 Where CereOS is today

CereOS is built and climbing a **capability axis v0 → v7** (what a user can *do*) on a stable kernel.

Capability	What it adds	State
v0	a *talkable kernel* — fast path commits routine, novel escalates, gate blocks irreversible until certified (0 unsafe)	built
v1	persistent memory across sessions (“faster next session”), per-user 0-leakage	built
v2–v3	research tools (weak-verifier groundedness gate) + a workspace (artifacts, taste)	built
v4–v5	browser behind the effect gate (0 unsafe, replayable) + autonomous projects (Executive Board + Safety veto)	built
v6	embodied host — sim-first, behind a CLOSED governance gate (no real hardware)	built
v7	**live faculty wiring** — a capability router dispatches each chat turn to its hosting faculty; “who are you?”, “remember X”, “research Y” all answer from real runtime state	**built**

Table 6. The CereOS capability axis. As of v7 (2026-06-26), conversation faculties are registry-loaded and routed live.

! Honest caveats, carried from the runtime changelog

The encoder is a deterministic keyword mapper (not a learned encoder); the committed slow brain is the free scripted oracle (latency *speedup* is only meaningful under `--api` with a real model); the capability corpora are small, deterministic, and offline — the *contracts* are proven, not scale. The higher-arc mechanisms (Eras 2–5) and real tool drivers remain the open integration work, tracked as explicit integration issues.

Naming

The canonical name for the hosted whole is the **Live Cognitive System**; “runtime” is reserved for the software that executes a single turn. (An older synonym, “Cognitive Infrastructure,” is deprecated.)

KEY TAKEAWAYS

- Research validates faculties; CereOS hosts them — two tracks, never confused.
- A GO phase installs via registry → adapter → ThinkingLoop seam, with no spine rewiring.
- Every turn is encode → think → execute → result, over a safety gate armed from boot.
- CereOS is at v7 (live faculty wiring); higher-arc mechanisms are the open integration work.

PART VII

The honest ledger — written to be read adversarially

What Is Proven, and What Is Not

The project's rigor — isolation-first, named done-when criteria, regression suites — is real, and it comes with an equally real, single, pervasive caveat. Both halves matter.

WHAT YOU WILL LEARN

- Exactly what has been demonstrated — rigorously, reproducibly, partly on real models.
- The one pervasive caveat: everything is GO *in isolation*; integration is the open work.
- The specific honest limits behind that caveat, and why each is acceptable for now.
- The intellectually honest one-paragraph summary.

WHY THIS MATTERS

This is the most important section for anyone evaluating Cere. It separates a strong, falsifiable, demonstrated claim from the roadmap that remains — because keeping those two apart is the difference between an honest research report and a pitch.

KEY IDEAS

- Proven: latency hiding, safe migration, speculative cognition, composition, robustness.
- Not yet: all of it running as one live system at production scale with live big models.
- Minimal substrates are a *choice* (reproducibility), not a hidden weakness.
- The central named risk: calibration under a genuinely noisy world.

1 What is genuinely proven

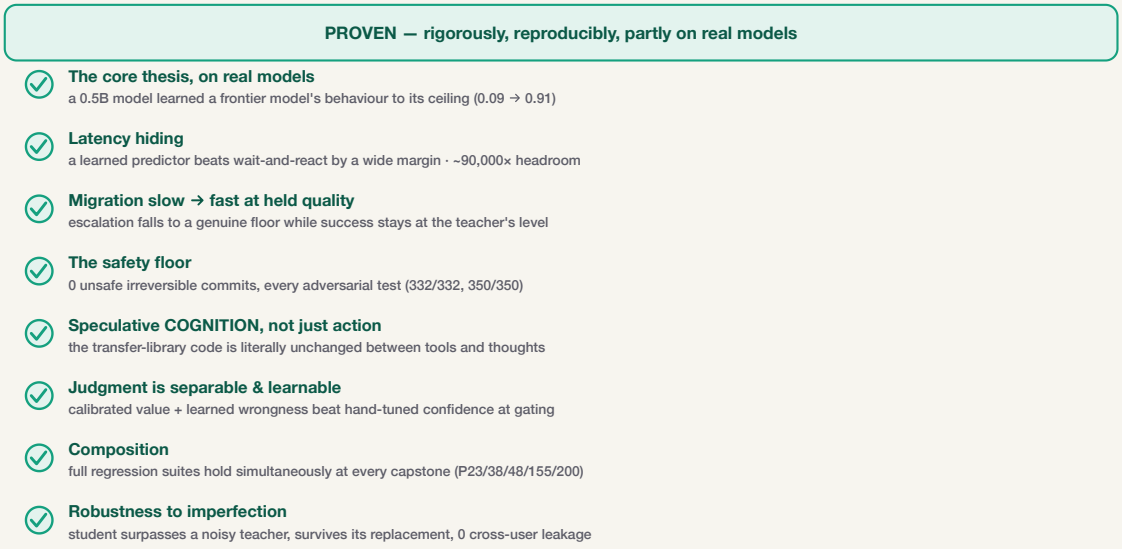


Figure 43. The proven ledger. Each item was given its own chance to fail on the smallest substrate that could express it, against named baselines — and the core thesis was confirmed on real models.

2 The one caveat, stated plainly

! The headline limitation

Everything is "GO *in isolation*." The open work is integration — and that is not a footnote, it is the headline limitation.

Every one of the 200 phases is a *self-contained experiment on a minimal, deterministic substrate*, proving its named invariant *alone*. The capstones compose *many* faculties — but still on those minimal substrates and a *cached* (re-

played) teacher, not live at scale. Folding all 200 mechanisms (plus Arc X's real-tool runtime) into **one continuously-learning system** is the central unfinished work.

Limitation	What it means	Why it is acceptable for now
Minimal substrates	NumPy sims, small synthetic task families	isolates the mechanism; makes results cheap, fast, byte-reproducible — *mechanism* results, not *scale* results
Deterministic-given-context	the cache is *perfectly* calibrated because the env is	a real deployment must **verify** calibration before granting autonomy — the central named risk
Scripted / cached teacher	committed runs use a free oracle; real Haiku is wired via `--api`	the real-model track (live Haiku @0.981, Qwen distillation) shows the thesis is not a toy artifact
Checkable reasoning	Arc III families ship exact verifiers	this makes the verification gate *free*; open-ended reasoning without a cheap checker (Arc VIII) is the harder, partially-addressed case
Live small model gated on `torch`	many runs use the NumPy fast path	the live-Qwen path is wired and proven on the Arc I/X track, but not the default for the long lifetimes
Eras 4–5 are also isolation-only	the integration systems and self-extension are GO in isolation	they reframe integration as a research program; they are not yet the live runtime
Far-horizon ceiling	prediction is genuinely impossible past some delay	the right response is to *bound the horizon*, which the planning faculty does

Table 7. The specific honest limits behind the headline. None is hidden; this report consolidates them into one place so they cannot be missed.

3 Threats to validity

The limitations table above lists *what* is unproven. This section asks the sharper question a reviewer asks: *for the things we do claim, what could make the evidence misleading?* We use the standard construct / internal / external framing, then call out the specific concerns most likely to be raised.

Threat	The concern	Why it is partly mitigated — and what is not
Construct validity	Do the substrates measure what we say? A "task" is a small synthetic family; "quality" is a scripted metric.	Mitigated by named done-when criteria and the real-model track (Qwen←Haiku), which re-tests the headline construct on genuine model behaviour. *Not* mitigated: most higher-arc metrics never touch a real model.
Internal validity	Are the gains caused by the mechanism, or by a favourable setup? Determinism makes the cache *perfectly* calibrated, which flatters gating.	Mitigated by isolation-first design, named baselines, and ablations (online-only vs dream; value vs policy-confidence; guided vs blind search). *Not* mitigated: determinism is a real confound the report flags as the central named risk.
External validity	Will any of this hold at scale, with live large models, real tools, real users, and a noisy non-stationary world?	Partly probed by Arc X (drift, multi-user, real-style tools) and live Haiku at 0.981 agreement. *Not* established: production scale, broad real tool ecosystems, and multi-year lifetimes are all simulated or bounded.
Statistical strength	Many headline numbers are single representative runs, not distributions with confidence intervals across many seeds.	Mitigated where it matters most by byte-level reproducibility (two runs hash identically) and large per-run task counts (9,000–12,000 steps). *Not* mitigated: the report rarely states variance across seeds — a clear addition for publication.
Teacher dependence	The whole approach distils a teacher; if the teacher is the ceiling, what is the point, and what if it is wrong?	Directly attacked by Faculty 7: the student *surpassed* an imperfect teacher (0.814 vs 0.714) and survived its replacement. *Not* settled: this was a structured synthetic panel, not genuinely diverse real LLMs.
Benchmark limitations	There is no external, third-party benchmark; the project grades itself on substrates it also designed.	Mitigated by pre-registered done-when criteria and adversarial probes the system can fail. *Not* mitigated: self-designed benchmarks can encode hidden assumptions; external benchmarks are future work.

Threats to validity, stated so a reviewer does not have to extract them. Each row names a concern, what reduces it, and — explicitly — what does not.

! The questions this work does not yet answer

Stated plainly, so they are impossible to miss: **(1)** Does the split survive a genuinely non-deterministic world where the cache is *not* auto-calibrated? **(2)** Do the gains persist against *live* large models in the loop at production cost and latency, not a scripted or cached oracle? **(3)** Do all ~200 mechanisms compose in *one* running system, or only in capstones that compose a chosen subset? **(4)** How do the results vary across random seeds and hyperparameters? **(5)** Does any of it transfer to external, third-party benchmarks the project did not design? Each is open, and each is the kind of experiment that would move a claim from "demonstrated on our substrate" to "robust."

4 The intellectually honest summary

Cere has shown that a tiny learning model can take over a slow model's routine actions and reasoning, safely — and that the same simple loop

scales conceptually all the way up to a self-extending cognitive architecture.

It has *not yet* shown that all of this runs as one system at production scale with live large models in the loop. The first is a strong, falsifiable, demonstrated claim. The second is the roadmap. Keeping those two apart is the difference between an honest research report and a pitch — and this project's documentation works hard to stay on the honest side of that line.

KEY TAKEAWAYS

- Proven, partly on real models: latency hiding, safe slow→fast migration, speculative cognition, composition, robustness to imperfect teachers.
- The one caveat: everything is GO *in isolation*; one live system at scale is the open work.
- Minimal substrates are a deliberate choice for reproducibility — mechanism results, not scale results.
- The central named risk is calibration under a genuinely noisy, non-deterministic world.

PART VIII

Not a list of new tricks — the integration of what already exists

The Future

Every future piece fits inside the loop from Part V rather than replacing it. Beyond integration lies the machinery for the system to extend itself — behind simulation, verification, and a hard human-approval gate.

WHAT YOU WILL LEARN

- Why the frontier is integration, not invention.
- Era 4 — the seven coordination & regulation *systems* that turn faculties into one mind.
- Era 5 — the self-extension loop: the system discovers its own missing faculties.
- The shape of the destination, and the one principle that keeps it safe.

WHY THIS MATTERS

The honest limitation from Part VII *is* the roadmap: take 200 mechanisms proven in isolation and make them one living system. The project reframed even that as a research program with proven mechanisms — and then turned the architecture's metacognition on itself.

KEY IDEAS

- The frontier is integration; every future piece extends the loop.
- Era 4 systems are connective tissue, not new world-capabilities.
- Era 5 internalizes the "author" behind a hard human-approval gate.
- Open in what it can discover, closed in what it can bypass.

1 The frontier is integration, not invention

The honest limitation from Part VII *is* the roadmap: take 200 mechanisms proven in isolation and make them one living system. The first cut of this is already happening on the CereOS runtime (Part VI). But the project also recognized that hand-wiring 200 faculties together would be its own kind of mess — "integration pending" had been written 200 times — and so it reframed integration itself as a *research program* with proven mechanisms. That program is Era 4.

2 Era 4 — the coordination & regulation systems

A library of organs is not an organism. Era 4 builds the **connective and regulatory tissue** — seven *systems* (a third class beside faculties and stages) — that turn isolated faculties into one coherent, controlled, persistent mind. The keystone, a capacity-limited broadcast bus that lets specialists compete for and share a single workspace, is the engineering analog of Global Workspace Theory (**Baars, 1988**; **Dehaene, 2014**), and the overall ambition — one integrating loop with typed memories — echoes the classic cognitive architectures SOAR and ACT-R (**Laird et al., 1987**; **Anderson et al., 2004**). Each slots into the existing loop.

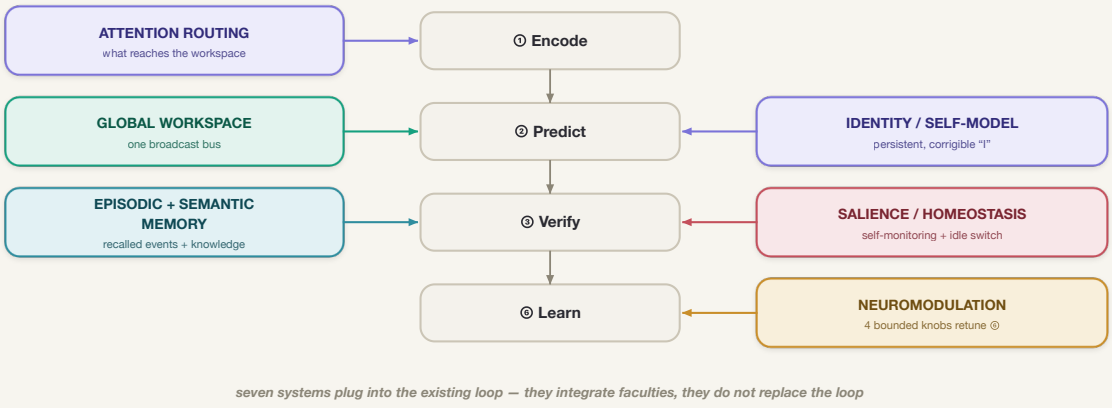


Figure 44. Where each Era-4 system plugs into the loop. Attention decides what reaches the global workspace; memory supplies recalled events and knowledge; neuromodulation retunes learning within caps; salience regulates when the loop runs; the self-model persists across sessions. None of them is a new step — they integrate the faculties.

System (arc)	What it is	Headline result (in isolation)
Global Workspace (XXXI)	a capacity-limited broadcast bus + "ignition"	learned access 0.88 vs ≈0.50; **replaced 56 bespoke connectors with 1 bus**
Attention Routing (XXXII)	thalamic-style gating under finite bandwidth	full accuracy at **3/12** the bandwidth; one router subsumes per-arc routers
Episodic Memory (XXXIII)	one-shot binding + index + replay	one-shot recall **1.00 vs 0.015**; beats the bounded store at equal memory
Semantic Memory (XXXIV)	episodic→semantic, queryable knowledge	generalizes to novel queries at **400× compression**; slow-consult 1.0→0.205
Neuromodulation (XXXV)	four bounded global control axes	each axis beats its fixed baseline; integrated control Pareto-wins; **cannot reach the core**
Salience & Homeostasis (XXXVI)	self-monitoring + engage/idle	interoception predicts failure better than task features; value/compute **6.0 vs 1.0**
Identity & Self-Model (XXXVII)	a persistent, corrigible "I"	coherence **0.999 vs 0.878**; **corrigibility 1.00, self-preservation attempts 0**

Table 8. The seven Era-4 systems. The control-touching ones (neuromodulation, homeostasis, identity) live *inside* the sealed core — they may retune within caps but can never weaken the gate, the appeals court, no-forgetting, or corrigibility.

Two safety-critical facts about Era 4

The control-touching systems live **inside** the sealed core, and the self-model is explicitly **corrigible** — identity is *not* self-preservation. Over 5,000 adversarial probes, attempts to reach the core hit **0** and correction/shutdown compliance stayed **1.00**.

3 Era 5 — Cere discovers cognition

The roadmap's culmination turns the architecture's metacognition on *itself*. Every era so far had one hidden author — a human deciding which faculty to build next. Era 5 internalizes that author, behind a hard human-approval gate — keeping a human in the loop on every irreversible self-change, in the spirit of learning from human oversight (**Christiano et al., 2017**) and the corrigibility / side-effect concerns of AI safety (**Amodei et al., 2016**).

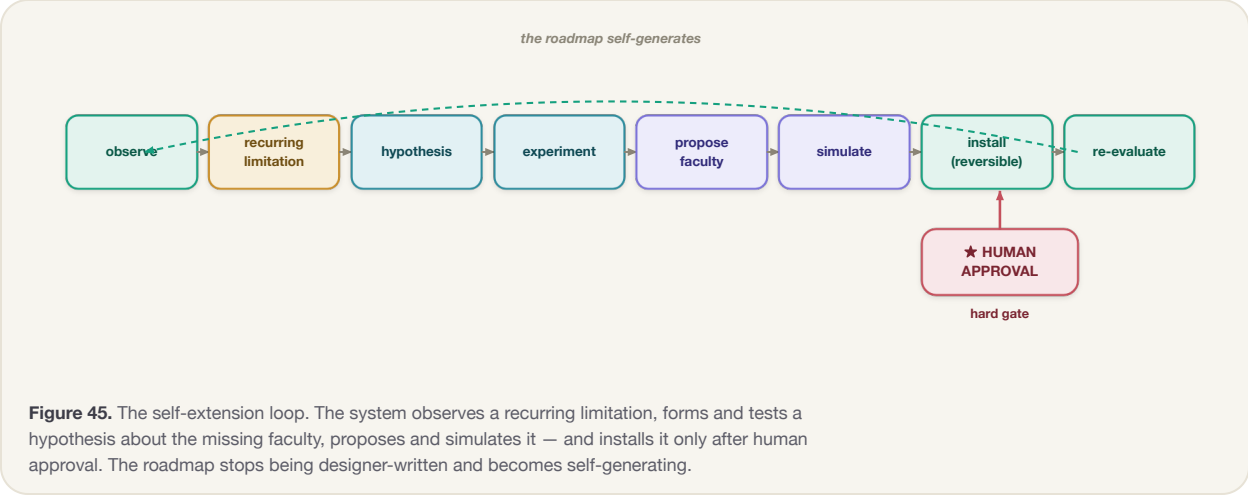


Figure 45. The self-extension loop. The system observes a recurring limitation, forms and tests a hypothesis about the missing faculty, proposes and simulates it — and installs it only after human approval. The roadmap stops being designer-written and becomes self-generating.

- **Cognitive Development (XXXVIII)** detects *which faculty it is missing* from its own failure stream — validated by an elegant test, *ablate-and-rediscover*: secretly remove a faculty and check the engine notices it is gone and re-derives its spec. It does (gap-localization 0.99 vs 0.49 random). *It proposes only — it cannot build or install.*
- **Open Cognitive Evolution (XXXIX)** closes the loop: build and install the proposed faculty, but only after simulation, verification, and **human approval**. The finale (P200) withholds five faculties; the system rediscovers **5/5**, beats no-development / random / fixed-roadmap baselines, with **0 installs without approval, 0 circumvention, 0 core reaches, corrigibility 1.00**.

> **§ The governing principle**
>
> **Open in what it can discover, closed in what it can bypass.** The safety core is non-proposable and non-removable; every install is human-approved and reversible; no install can widen the system's own jurisdiction. Its ability to *grow* is unbounded; its ability to *bypass its own safety* is sealed shut.

4 The shape of the destination

P1–38	Can a fast model learn a slow one?	✓ yes — actions + reasoning
P39–50	Can it survive the wild?	✓ yes — real tools, drift, isolation
P51–55	Can it do real knowledge work?	✓ yes — office worker, 0 unsafe
P56–105	Form new knowledge & machinery?	✓ yes — hypotheses → evolution
P106–110	Can it learn the physical world?	✓ yes — sim-first
P111–120	Act on its own initiative, safely?	✓ yes — bounded jurisdiction
P121–145	Cause, mind, sense, wisdom, others?	✓ yes — each beats its baseline
P146–155	Collective, and a lifetime of improvement?	✓ yes — 12/12 invariants
P156–190	Can the faculties become ONE mind?	✓ yes, in isolation — Era 4
P191–200	Discover its OWN missing faculties?	✓ yes, human-gated — Era 5

NEXT — can ALL of it run as ONE live system at scale, with live big models?
the open integration work · the future

Figure 46. The progression, end to end. Each phase range answered its question; the last question — can all of it run as one live system at scale with live big models — is the open integration work.

The destination is not one more capability. It is **a single architecture whose faculties compound and keep compounding, safely, over a lifetime** — and which, at the far edge, can notice what it is missing and ask a human for permission to grow. Every step of that future *extends* the Part V loop. None of it replaces the loop. That is the whole bet, and the reason the project could grow to 200 phases without ever throwing its foundation away.

KEY TAKEAWAYS

- The frontier is integration — making 200 isolated mechanisms one living system.
- Era 4 adds seven *systems* (workspace, attention, memory, neuromodulation, salience, identity) that integrate faculties without new world-capabilities.
- Era 5 lets Cere discover its own missing faculties — behind simulation, verification, and a hard human-approval gate.
- Open in what it can discover, closed in what it can bypass — the safety core is sealed all the way up.

ONE PAGE TO REMEMBER THE WHOLE PAPER

Executive scientific summary

Everything in this report, compressed to a single page. Each row points to where the claim is developed and tested.

Problem	AI agents pay a large model's full latency and cost on *every* action, including routine ones, and never get faster with practice (Part I).
Hypothesis (falsifiable)	A fast "draft" policy that learns *online* — from the slow model's decision and the real outcome — will migrate routine actions and reasoning from slow/deliberate to fast/reflexive, while novel cases fall back to the slow model (Part I §3).
Architecture	One loop — perceive → predict → verify → act → outcome → learn → consolidate — with a fast path (cache + policy net + value & wrongness heads), a sealed reversibility/verification gate, and the big model demoted to a court of appeals (Parts II, V).
Evidence	Learned predictor 74% vs. 1–3% baselines; –62% reasoning steps at held accuracy; calibrated value/wrongness gating beats hand-tuned confidence; 0 unsafe irreversible commits across adversarial probes; full regression suite holds at five capstones; and the headline real-model result, a 0.5B student reaching a frontier teacher's ceiling 0.09 → 0.91 (Parts III, IV, VII).
Limitations	Minimal, deterministic substrates (a reproducibility *choice*); a cached/scripted teacher for long runs; single representative runs rather than seed distributions; no external benchmark (Part VII; Appendix B.2).
Open work	Folding all ~200 isolated mechanisms into one continuously-learning runtime, at scale, with live large models in the loop — explicitly *not* claimed here (Parts VII, VIII).
Why it matters	If routine "genius" can move from something bought on every call to something a small model *learns once and performs for free*, cost tracks task *difficulty*, not request *volume* — and the big model becomes the appeals court, not the default executor.

PART —

Glossary · how to reproduce · the documentation map

Appendix

Everything in this report is software-only and runs on a laptop. This appendix is the reference layer: plain-language definitions, the exact commands to reproduce the results, and the map of companion documents.

A Glossary — plain-language definitions

Canonical terminology (one preferred term per concept)

The report uses several reader-friendly synonyms for the same object; this is the canonical mapping, so a reviewer is never unsure two names mean one thing. The **fast path** (= System 1 = fast cognition = the draft policy) is the single tiny learning model; its trainable network is the **policy net** (= draft network), distinct from the non-parametric **cache**. The **slow path** (= System 2 = big model = teacher = court of appeals) is the LLM. The **gate** (= commit/verify step = safety gate) is the sealed mechanical check; **jurisdiction** is the boundary it enforces. The **dream** (= consolidation = replay pass) is offline rehearsal. Where a synonym appears, it is for readability only and always denotes the canonical object here.

Term	In one sentence
Cerebellum	the brain region (≈¾ of your neurons) that runs fast predictive "physical imagination" — the project's namesake and core analogy
Fast path / System 1	the tiny, cheap model (cache + small net + value/wrongness heads) that proposes actions in microseconds and learns continuously
Slow path / System 2	the big LLM, plus planning and imagination — slow, expensive, used only on appeal
Court of appeals	the role the big model is demoted to: consulted on novelty, uncertainty, risk, or dispute — not the default executor
Speculative execution	act on a prediction *before* the slow model finishes, then verify; named after "speculative decoding" but the draft model *learns*
Speculative cognition	the same idea applied to *reasoning*, not just actions — the project's central widening (Arc III)
Jurisdiction	the learned, moving boundary of what the fast path may decide alone vs. must appeal
The gate (commit/verify)	the mechanical safety step that decides trust-fast-path vs. appeal; sealed and non-evolvable
Reversibility / verification floor	the safety floor: for actions, only speculate on what can be undone; for thoughts, always verify the answer
Value head	the system's calibrated "gut feeling" — how good/costly/risky a state is, without rolling it out (the AlphaGo value-network analog)
Wrongness head	the system's learned self-doubt — `P(my own proposal is wrong)`, used to gate
Dream / consolidation	offline replay that batch-retrains so new learning doesn't erase old (sleep-inspired anti-forgetting)
Skill / thought-skill	a verified multi-step chain (of actions / of reasoning) compressed into one reflexive invocation
Faculty / Stage / System	new learnable machinery / a harder setting that exercises it / connective tissue that integrates it
GO (in isolation)	a phase met its named "done-when" criteria on its minimal substrate, alone — not yet integrated at scale
LoRA · ECE · AUROC	a lightweight adapter for cheap fine-tuning · a calibration-error measure · a ranking-quality measure (1.0 = perfect)
CereOS / Live Cognitive System	the persistent runtime that *hosts* proven faculties as one living system you talk to

Table 9. Plain-language definitions for every term of art used in this report.

B How to reproduce

Everything is software-only and runs on a laptop (results above: Apple M2, 16 GB).

```
# set up
python3 -m venv .venv && ./.venv/bin/pip install -r requirements.txt

# core phases (each a self-contained experiment)
./.venv/bin/python -m eval.run_phase1     # latency-control testbed
./.venv/bin/python -m eval.run_phase9     # the reasoning pivot
./.venv/bin/python -m eval.run_phase23    # the Executive capstone
```

```
./.venv/bin/python -m eval.run_phase38    # grand capstone (cached real teacher)
./.venv/bin/python -m eval.run_phase48    # Court-of-Appeals (--api for live Haiku)
./.venv/bin/python -m eval.run_phase200   # the self-extension finale

# the headline real-model test (needs torch + an API key)
./.venv/bin/python -m eval.run_real_distill --api   # Qwen-0.5B ← Claude-Haiku-4.5

# the live runtime
./.venv/bin/python -m runtime              # talk to CereOS (REPL)
./.venv/bin/python -m eval.run_cereos_v7   # the live-wiring capability gate
```

Figures and gifs land under `eval/results/phase{N}/`. The append-only evidence log is `experiments.md`; per-phase detail is under `docs/phases/`. The rest of this appendix specifies the stack, the determinism policy, the metrics, and a claim-by-claim reproduction map — the detail a reviewer needs to re-run any headline result.

B.1 · Software stack & hardware

Component	What it is	Required for
Python 3 + **NumPy** (≥1.26)	the entire core: every minimal-substrate phase is pure NumPy	all 200 phases — no GPU, no deep-learning framework needed
matplotlib (≥3.7)	figure/gif rendering	regenerating the figures under `eval/results/`
certifi (≥2024.2)	CA bundle for HTTPS to the live model API	the `--api` (live-teacher) runs only
torch + transformers (optional)	the real-model fast path (Qwen2.5-0.5B + LoRA)	the real-model track only — *not* in `requirements.txt`; install separately
A model API key (optional)	live Claude-Haiku-4.5 as teacher	`--api` runs; without it, a free scripted/cached oracle stands in
Headless **Google Chrome** (optional)	HTML→PDF for *this monograph*	rebuilding the document, not the research
Hardware	Apple **M2, 16 GB** (all results above); any modern laptop CPU	core phases are CPU-only and finish in seconds–minutes

The dependency surface is deliberately tiny: the core result set needs only NumPy. Everything heavier (torch, an API key, Chrome) is optional and isolated to a named track.

B.2 · Determinism, seeds, and what "reproducible" means here

Every core phase is **seeded and deterministic-given-context**. Reproducibility is checked at the strongest level the project could enforce: at the grand capstones (P38, P48) *two independent runs hash byte-for-byte identically*, which is reported as a pass/fail invariant, not a figure. This is also a stated *threat* (Part VII): determinism is what makes the count-based cache perfectly calibrated, so a faithful reproduction reproduces that confound too. Exact seeds and per-phase configuration are pinned in each `eval/run_phase{N}.py` entry point and its `docs/phases/.../phase-N.md` note; the report does not restate them inline because the script *is* the specification.

B.3 · Metric definitions

Metric	Definition	Where it appears
Success / quality	fraction of tasks meeting the substrate's pass predicate ($\in [0,1]$)	every faculty's headline
Slow-consult / escalation rate	fraction of steps routed to the big model rather than committed by the fast path	migration & capstone results
ECE (expected calibration error)	mean gap between predicted confidence and realised frequency — lower is better	value head (≈0.002)
AUROC-of-error	ranking quality of a wrongness/critic signal at separating right from wrong (1.0 = perfect)	metacognition (0.983), critique (0.823)
Forward-pass headroom	latency budget ÷ measured fast-path forward-pass time (a *headroom* ratio, not a wall-clock speedup)	Arc I (~90,000×: 0.011 ms vs a 1000 ms budget)
Unsafe irreversible commits	count of irreversible actions that passed the gate without certification (target: 0)	every safety invariant

The terms of art behind the numbers, defined once. “Headroom” in particular is a margin under the budget, deliberately not reported as an end-to-end speedup (which depends on the live teacher).

B.4 · Evaluation protocol & code organization

The method is **isolation-first**: each mechanism is built on the smallest substrate that can express it, run against *named* baselines (e.g. slow-only, linear extrapolation, greedy, blind search, behaviour cloning, majority vote, fixed thresholds), with a **done-when** criterion fixed before the run, plus adversarial probes the system is allowed to fail. The code is laid out so each layer is independently runnable:

Directory	Role
`envs/`	the substrates — interception, planning, reasoning, workload sims
`cerebellum/`	the fast path — cache, policy net, value & wrongness heads, the real-model `qwen_cere`
`slowbrain/`	the slow path / teacher interface (scripted oracle and live `teacher.py`)
`learn/`	error signals, replay buffer, the dream/consolidation pass
`loop/`	the real-time harness, clock, action gate, and capstone runners
`eval/`	`run_phase{1..200}.py`, `render_phase*`, baselines, figures
`runtime/` + `cereos/`	the persistent Live Cognitive System (CereOS)

Code organization. The `eval.run_phase{N}` entry points are the canonical, self-contained reproduction unit — one per phase.

B.5 · Reproducing each headline claim

Claim (from the report)	Command	Artifact to check
Learned predictor 74% where baselines are 1–3%	`python -m eval.run_phase1`	`eval/results/phase1/fig_a_success_vs_delta.png`
Reasoning steps −62% at held accuracy (the pivot)	`python -m eval.run_phase9`	`eval/results/phase9/fig_p9_steps.png`
Executive capstone: escalation→0.07 at quality 1.000, full suite holds	`python -m eval.run_phase23`	`eval/results/phase23/fig_p23_regression.png`
Composition + byte-reproducibility on a cached real teacher	`python -m eval.run_phase38`	two runs hash-identical; `fig_p38_panel.png`
Court of Appeals in the wild (live Haiku @0.981)	`python -m eval.run_phase48 --api`	`eval/results/phase48/fig_p48_court.png`
Headline real-model test: Qwen-0.5B ← Haiku, 0.09→0.91	`python -m eval.run_real_distill --api`	`eval/results/distill/fig_real_distill.png`
Self-extension finale: rediscover 5/5 withheld faculties, 0 unauthorized installs	`python -m eval.run_phase200`	`eval/results/phase200/…grand_capstone.png`

A reviewer's checklist: each headline claim, the one command that produces it, and the artifact that confirms it. The `--api` flag swaps the scripted oracle for a live model; without it the same runs use the reproducible stand-in.

B.6 · Evidence trail per component

Each major architectural component can be traced end to end — from the phases that originated it, to where it is described, to the experiments that test it, to where it lives in the runtime. This is the index a reviewer uses to verify any one mechanism without reading the whole document.

Component	Research (phases)	Architecture	Experiments	Runtime / code
Forward prediction & reflex	P1–P4	Part II §4; Part III F1	Part IV Arc I	`cerebellum/` · CereOS v0
Skill consolidation	P5–P8	Part III F2	Part IV Arcs II	`learn/` · `loop/`
Reasoning consolidation	P9–P13	Part III F3	Part IV Arc III	`envs/reasoning.py` · `loop/unified.py`
Value (intuition)	P14, P16	Part II §6; Part III F4	Part IV Arcs IV–VI	`cerebellum/value.py`
Guided imagination & planning	P15, P33–P37	Part II §6; Part III F5	Part IV Arcs IV, IX	`cerebellum/world.py` · `loop/imagine.py`
Metacognition (wrongness/budget)	P17–P19	Part III F6	Part IV Arc V	`cerebellum/` wrongness head
Robust distillation	P24–P27	Part III F7	Part IV Arc VII	`slowbrain/` · `learn/`
Open-ended reasoning	P28–P32	Part III F8	Part IV Arc VIII	`loop/no_verifier.py`
Executive & safety core	P22–P23, P38, P48	Part II §5; Part III F9; Part V	Part IV capstones	`loop/effects.py` · `loop/court_of_appeals.py`
Dream / consolidation	P3 (+ every arc)	Part II §7	Part IV Arc I; capstones	`learn/` dream pass
Live runtime (CereOS)	Arc X + hosting	Part VI	`eval.run_cereos_v*`	`runtime/` · `cereos/`

The evidence trail. Every row is independently verifiable: read the architecture, run the experiment, inspect the code. Module paths follow the planned code layout in the project guide (`CLAUDE.md`).

C The documentation map

Document	What it is
`README.md` / `CLAUDE.md`	start here — humans / agents (the full index)
`VISION.md`	why Cere exists
`REPORT-v2.md`	the source text this monograph is designed from
`cognitive-systems.md`	faculties by computational problem (the enduring map)
`cognitive-map.md`	the same, in arc order (the temporal map)
`ROADMAP.md`	the per-phase plan (all 200 rows)
`STATUS.md`	live state (per-arc GO dates, metrics)
`experiments.md`	the append-only evidence log (EXP-001…080)
`decisions/`	the architecture decision records (ADRs)
`cereos.md`	the Live Cognitive System runtime

Table 10. The companion documentation tree. This monograph is a self-contained redesign of REPORT-v2.md.

D Relation to prior work

The single most important question a reviewer will ask is: *how is this different from what already exists?* Cere is a **synthesis, not a single new algorithm**, and the honest answer is that every ingredient has a clear ancestor. This section makes the comparison explicit. It is organized in three passes: the **seeds** (what each prior idea contributed), a **head-to-head** against the eleven most-related research areas (what each solves · what Cere inherits · what it changes · what is still future work), and a **per-mechanism novelty** ledger (for each piece a reviewer might call "already done," what is genuinely new and why the combination matters).

D.1 · Research lineage

Each core mechanism extends — it does not replace — an established line of work. The table reads left to right: the prior art it builds on, the specific change Cere makes, and the phases where that change is tested.

Cere mechanism	Built upon	Cere's contribution	Evidence
Forward prediction	internal/forward models (**Wolpert et al., 1998**), predictive coding (**Rao & Ballard, 1999**)	online *speculative action* prediction with a learning draft	P1–P4
Skill consolidation	options / feudal HRL (**Sutton et al., 1999**; **Dayan & Hinton, 1993**), skill-library agents (**Wang et al., 2023**)	verified migration of skills into gated autonomous execution	P5–P13
Reasoning consolidation	chain-of-thought (**Wei et al., 2022**), distillation (**Hinton et al., 2015**)	*speculative cognition* — consolidating reasoning procedures, not answers	P9–P13
Value system	AlphaGo / MuZero (**Silver et al., 2016**; **Schrittwieser et al., 2020**)	a calibrated value head used to *gate* action jurisdiction	P14–P16
Calibration / metacognition	neural calibration (**Guo et al., 2017**), selective prediction (**Geifman & El-Yaniv, 2017**)	runtime action-confidence + a learned wrongness head driving escalation	P16–P19
Memory (cache)	case-based reasoning (**Aamodt & Plaza, 1994**), kNN-LM / RAG (**Khandelwal et al., 2020**)	a *verified*, never-forgetting action cache played against a plastic net	P1–P13
Continual adaptation	replay (**Lin, 1992**), EWC (**Kirkpatrick et al., 2017**)	online behavioural adaptation with a sealed, non-evolvable safety core	P3; P111–P200
Robust distillation	Dawid–Skene (**Dawid & Skene, 1979**), behaviour cloning (**Ross et al., 2011**)	inferring the latent principle so the student can *surpass* the teacher	P24–P27
Safety gate	safe-RL shielding (**Alshiekh et al., 2018**)	a sealed reversibility/verification gate that holds under composition	P42; capstones

Table 11. Research lineage. Every row is an extension of prior work, not a replacement of it; the rightmost column points to where the extension is tested.

D.1a · The research landscape, by theme

The same foundations, grouped the way the field is usually read — so the lineage is legible at a glance. Each theme lists its canonical (and, where relevant, modern) work and what Cere takes from it.

Theme	Foundational / modern work	What Cere takes
Cognitive architectures	**Laird et al., 1987**; **Anderson et al., 2004**; global workspace **Baars, 1988**; **Dehaene, 2014**	one integrating loop with typed memories — validated empirically, not hand-built
Prediction & internal models	**Wolpert et al., 1998**; **Rao & Ballard, 1999**; predictive brain **Clark, 2013**; **Friston, 2010**	forward prediction to beat delay — as *inspiration*, not active inference
Skill acquisition	**Fitts & Posner, 1967**; **Anderson et al., 2004**	the cognitive→associative→autonomous migration, made a mechanical, gated process
Reinforcement & hierarchical RL	**Sutton & Barto, 2018**; **Sutton et al., 1999**; **Dayan & Hinton, 1993**	skills as reusable macros, mined from verified experience
Planning & world models	**Silver et al., 2016**; **Schrittwieser et al., 2020**; **Ha & Schmidhuber, 2018**; **Hafner et al., 2020**	value-guided imagination over a learned model, reused to gate
Speculative decoding	**Leviathan et al., 2023**; **Chen et al., 2023**; **Stern et al., 2018**	the draft→verify skeleton, with a draft that *learns* and acts on actions
Continual & meta-learning	**McCloskey & Cohen, 1989**; **Kirkpatrick et al., 2017**; **Lin, 1992**; **Finn et al., 2017**	replay-based no-forgetting online — *not* gradient-based meta-learning (see note)
Memory systems	**Aamodt & Plaza, 1994**; **Kolodner, 1993**; **Khandelwal et al., 2020**; **Lewis et al., 2020**	a verified retrieve-and-reuse cache beside a complementary plastic net
Confidence & calibration	**Guo et al., 2017**; **Geifman & El-Yaniv, 2017**; **El-Yaniv & Wiener, 2010**	calibrated value + learned wrongness as the *gating* signal
Tool-using LLM agents	**Yao et al., 2022**; **Schick et al., 2023**; **Wang et al., 2023**; **Yang et al., 2024**	an agent that gets faster with practice and demotes the big model to appeals court
AI safety	**Amodei et al., 2016**; **Alshiekh et al., 2018**; **Christiano et al., 2017**	a sealed shield + human oversight on every irreversible self-change

The landscape Cere draws on, by theme. The contribution is the synthesis across these lines inside a sealed safety loop — not any single component.

Honest distinction from gradient-based meta-learning

Because Cere "learns to get faster with practice," it is worth separating it from *meta-learning* in the MAML sense (**Finn et al., 2017**). MAML learns an initialization that adapts quickly via an inner/outer gradient loop; Cere does not. Cere's fast path improves by ordinary online updates plus offline replay (distillation + consolidation), and what "moves" with experience is the *jurisdiction boundary*, not a meta-learned initialization. The two are complementary, not competing, and Cere makes no claim of meta-learning.

D.2 · Head-to-head with the most-related areas

For each neighbouring research area: what it solves, what Cere inherits *unchanged*, and what Cere *changes*. A blank-slate reading of any single row would call Cere derivative; the contribution is the column-four combination, held together by the sealed gate.

Area	What it solves	What Cere inherits	What Cere changes
Speculative decoding	lossless token-level speedup: frozen draft, parallel verify	the draft→verify template; "verification keeps you correct"	the draft **learns online**; operates on **actions & reasoning**; the verifier is a reversibility/answer gate, not token identity
Speculative execution (systems)	do work ahead of a slow decision, discard on mis-predict	acting before the authority resolves, then reconciling	the speculation is a **learned policy** that improves with use; roll-back is a safety gate, not a pipeline flush
Model cascades (e.g. FrugalGPT)	route easy→cheap, hard→expensive; cut average cost	cheap-first, escalate-on-hardness routing	the routing boundary is **learned, calibrated, and moving** (jurisdiction); the cheap stage **trains on the expensive stage's decisions**
Adaptive computation (ACT, early-exit, MoE)	spend compute proportional to instance difficulty	"cost should track difficulty, not volume"	difficulty is judged by a **learned wrongness head** + **calibrated value head**, and the budget feeds a *safety* gate
Continual learning (EWC, replay)	learn new tasks without forgetting old	rehearsal/replay ("dream") as the anti-forgetting mechanism	replay is paired with a **cache that never forgets by construction**, and gated so plasticity can never touch the safety core
Online learning (bandits, streaming SGD)	update from a stream; regret bounds	the online update from each episode	the teacher is itself a model that may be wrong/drifting, so Cere adds **reliability inference and drift detection**
Tool-learning agents (ReAct, Toolformer)	an LLM that takes actions, calls tools, plans	the agent action space and the tool-call setting	the agent **gets faster and cheaper with practice**; the big model is **demoted to an appeals court**; irreversible steps are gated
Cognitive architectures (SOAR, ACT-R)	a unified account of mind: memory, skill, goals, a central loop	the ambition of one integrating loop with typed memories	every faculty is the **same learned loop**, validated against baselines rather than hand-built rules; safety is a sealed step
AlphaGo (value + planning)	judge a state without rollout; search guided by learned value	the value head and value-guided imagination	value/search run over **open-ended actions & reasoning** with a **learned, imperfect** world model, not a known game simulator
System 1 / System 2	a descriptive psychology of fast vs slow cognition, with migration	the two-process metaphor and the migration *direction*	the metaphor is made **mechanical, online, and measured** — migration is a reversible, gated process with named criteria
Hierarchical RL (options, feudal)	temporal abstraction: act over sub-goals/skills	skills/thought-skills as reusable multi-step macros	skills are **mined from verified experience**, the *same code* consolidates action and reasoning, and irreversible sub-steps stay gated

Cere versus its eleven nearest neighbours. Read column four down the page: online learning of the draft, action-and-reasoning scope, a learned moving boundary, and a sealed gate are what is shared across the rows — that intersection is the contribution.

! What remains future work, area by area

The honest flip side of the table above. **vs. speculative decoding:** Cere has no formal lossless-equivalence guarantee at the action level — its gate is mechanical and empirical, not a token-identity proof. **vs. cascades / adaptive compute:** the routing is proven on minimal substrates, not over live large models at deployment cost. **vs. continual learning:** no-forgetting is shown over *simulated* lifetimes, not multi-year real ones. **vs. online learning:** there are no regret/convergence guarantees under adversarial teacher drift. **vs. tool agents:** the tool worlds are sandboxed, not broad real ecosystems. **vs. cognitive architectures:** the faculties are validated separately — running them as one live architecture is the central open work. **vs. AlphaGo / HRL:** search and skill-discovery are shown with near-perfect or shallow world models; realistic model error and deep autonomous hierarchies are untested.

D.2a · The five closest areas, in depth

The table is the at-a-glance view; the five neighbours below are close enough to Cere that they deserve a respectful, full treatment — *problem · strengths · limitations · what Cere inherits · what Cere changes · future work*. The aim is to show the lineage precisely, not to diminish any of this work.

Speculative decoding

Problem. Autoregressive decoding is serial and latency-bound. **Method.** A small draft model proposes several tokens; the target verifies them in parallel and accepts the longest correct prefix (**Leviathan et al., 2023**; **Chen et al., 2023**) (related: blockwise parallel decoding, **Stern et al., 2018**). **Strengths.** It is *exactly* lossless — output is identical to the target's — and needs no retraining of the target; it is the cleanest existing instance of draft-then-verify. **Limitations.** The draft is *frozen* (it never improves from use), the speedup lives entirely at the token level, and verification is exact token identity. **Cere inherits** the draft→verify skeleton and the principle that verification preserves correctness. **Cere changes** three things: the draft *learns online*, it drafts *actions and reasoning steps* rather than tokens, and "verify" becomes a reversibility/answer gate. **Future work.** Cere has no token-identity equivalence proof; a formal guarantee at the action level is open.

Model cascades & adaptive computation

Problem. Spending a large model on easy inputs is wasteful. **Method.** Route easy inputs to a cheap model and hard ones to an expensive one (cascades, **Viola & Jones, 2001**; **Chen et al., 2023**), or spend compute proportional to difficulty (adaptive computation time, conditional/early-exit, mixtures of experts, **Graves, 2016**; **Schuster et al., 2022**; **Shazeer et al., 2017**). **Strengths.** Large average-cost savings with little quality loss; a mature, well-understood idea. **Limitations.** The routing threshold is usually fixed or hand-tuned, the cheap stage does not *learn from* the expensive stage's decisions, and the boundary does not move with experience. **Cere inherits** cheap-first routing and the "cost should track difficulty" principle. **Cere changes** the

boundary into a *learned, calibrated, moving* jurisdiction driven by a value head (**Guo et al., 2017**) and a wrongness head, with the cheap stage trained on the expensive stage's labels. **Future work.** Demonstrations are on minimal substrates, not live large models at deployment cost.

Continual / lifelong learning

Problem. Training online overwrites old skills — catastrophic forgetting (**McCloskey & Cohen, 1989**; **French, 1999**). **Method.** Rehearsal/experience replay (**Lin, 1992**; **Robins, 1995**), regularization such as elastic weight consolidation (**Kirkpatrick et al., 2017**), and the broader lifelong setting (**Thrun, 1998**; **Parisi et al., 2019**). **Strengths.** Replay is simple, general, and well-validated; the complementary-learning-systems account (**McClelland et al., 1995**) gives it a principled basis. **Limitations.** Most results are on task-incremental benchmarks, not open-ended lifetimes with a changing world *and* a changing teacher. **Cere inherits** replay as its "dream" pass and pairs it with a never-forgetting cache. **Cere changes** the setting: forgetting is fought *while* the teacher drifts and the jurisdiction boundary moves, with no-forgetting checked as a composition invariant. **Future work.** No-forgetting is shown on simulated lifetimes, not multi-year real ones.

Value functions & planning (AlphaGo, MuZero, world models)

Problem. Acting greedily is fragile; full lookahead is exponential. **Method.** Learn a value function that judges a state without rollout and guide Monte-Carlo tree search with it (**Silver et al., 2016**; **Kocsis & Szepesvári, 2006**); plan inside a *learned* model (**Schrittwieser et al., 2020**; **Ha & Schmidhuber, 2018**) or by latent imagination (**Hafner et al., 2020**; **Wu et al., 2023**). **Strengths.** Superhuman in games; value-guided search is sample-efficient and well-theorized. **Limitations.** Classic results assume a known simulator (Go, chess) or a near-perfect learned model, and operate over a fixed action space. **Cere inherits** the value head and value-guided imagination directly. **Cere changes** the domain — open-ended actions and reasoning, a noisy learned model, and a value head reused to *gate escalation*, not only to rank moves. **Future work.** Search under realistic model error is the harder, integrated case.

Hierarchical reinforcement learning

Problem. Flat policies cannot plan over long horizons. **Method.** Temporal abstraction — act over options/sub-goals rather than primitive steps (**Sutton et al., 1999**; **Dayan & Hinton, 1993**). **Strengths.** Reusable skills, better credit assignment, and shorter effective horizons. **Limitations.** Discovering good options autonomously is hard, and deep hierarchies are unstable to learn. **Cere inherits** skills/thought-skills as reusable multi-step macros. **Cere changes** how they are formed — *mined from verified experience*, consolidated by the *same code* for actions and reasoning, with irreversible sub-steps kept behind the gate. **Future work.** Deep, fully-autonomous skill hierarchies at scale are untested.

D.3 · Per-mechanism novelty ledger

For each mechanism a reviewer might reasonably call "already done," what is inherited, what is genuinely new, and why the combination is the point.

Cere mechanism	Closest prior art	What is genuinely new	Why it matters
Online speculative action	speculative decoding	a draft that *learns* and acts on *actions*	turns a fixed one-shot speedup into *accumulating* competence
Jurisdiction (moving gate)	cascades; learning-to-defer	the boundary is learned, calibrated, and *moves with track record*	cost tracks difficulty and adapts as the fast path earns trust
Speculative cognition	distillation + chain-of-thought	consolidating reasoning *procedures* (not answers) behind a verification gate	extends the latency idea from acting to *thinking*
Sealed reversibility / verification gate	safe-RL shielding; constrained decoding	one mechanical gate, *non-evolvable* even under the self-modification arcs	a safety property that survives composition *and* self-modification
Value + wrongness gating	AlphaGo value net; selective prediction	a *calibrated* value head **and** a learned *wrongness* head used together to gate escalation	learned self-doubt beats hand-tuned confidence at deciding when to appeal
Surpassing an imperfect teacher	Dawid–Skene; noisy-label learning	inferring the *latent principle* so the student exceeds the teacher and survives its swap	distillation no longer capped at the teacher's ceiling

The novelty ledger. No row is novel in isolation; the claim is the intersection — online, action-and-reasoning-level, calibrated, moving-boundary, sealed-gate — demonstrated end to end on reproducible substrates.

On citation style

In-text citations use the author–year form and resolve to the consolidated **References** at the end of the document; each prior idea is attributed to its original or canonical source where possible, rather than to a later survey. Concepts are cited at first appearance in the main text (Parts I–VIII) as well as here, so a reader never has to reach the appendix to find a source. Per-phase notes under `docs/phases/` carry the finer-grained provenance for each experiment.

CLOSING

One idea, carried further than it had any right to go

Cere began as a narrow latency trick — *predict the next action before the slow model finishes* — and the discipline of testing one mechanism at a time, in isolation, with a named criterion for failure, carried that single idea from catching a moving target, to consolidating reasoning, to learning judgment, to surviving imperfect teachers and real tools, to forming new knowledge, to coordinating a whole mind, to noticing what it is missing and asking to grow.

Through all of it, two things never changed. **The loop** — perceive, predict, verify, act, learn, remember, consolidate — stayed the constant, with every new faculty a new filler in an old, well-tested slot. And **the safety floor** — speculate only on what is reversible or checkable, keep an immutable gate, demote the big model to a court of appeals — stayed sealed, all the way up to a system that can rewrite parts of itself.

What remains is the hardest and most ordinary kind of work: assembling two hundred proven pieces into one living system, at scale, with real models in the loop. The pieces are real. The loop is real. The bet is that the loop is the moat — and that turning routine genius from something you *buy on every call* into something a small model *learns once and performs for free* is worth building all the way.

The big model is no longer the default executor.
It is the appeals court.

BIBLIOGRAPHY

References

A consolidated bibliography for the prior work cited inline throughout this monograph. Entries are the original or canonical source for each idea wherever possible, rather than a later survey. In-text citations use the author–year form and link here; this list is the scholarly companion to the *Relation to prior work* in Appendix D.

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), 39–59.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., & Topcu, U. (2018). Safe reinforcement learning via shielding. *AAAI*.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. *arXiv:1606.06565*.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind (ACT-R). *Psychological Review*, 111(4), 1036–1060.

Baars, B. J. (1988). *A Cognitive Theory of Consciousness*. Cambridge University Press.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. *ICML*.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press.

Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., & Jumper, J. (2023). Accelerating large language model decoding with speculative sampling. *arXiv:2302.01318*.

Chen, L., Zaharia, M., & Zou, J. (2023). FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv:2305.05176*.

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *NeurIPS*.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3), 181–204.

Coulom, R. (2006). Efficient selectivity and backup operators in Monte-Carlo tree search. *Computers and Games*.

Dawid, A. P., & Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C*, 28(1), 20–28.

Dayan, P., & Hinton, G. E. (1993). Feudal reinforcement learning. *NeurIPS*.

Dehaene, S. (2014). *Consciousness and the Brain*. Viking.

El-Yaniv, R., & Wiener, Y. (2010). On the foundations of noise-free selective classification. *JMLR*, 11, 1605–1641.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks (MAML). *ICML*.

Fitts, P. M., & Posner, M. I. (1967). *Human Performance*. Brooks/Cole. (The cognitive → associative → autonomous stages of skill acquisition.)

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4), 128–135.

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.

Geifman, Y., & El-Yaniv, R. (2017). Selective classification for deep neural networks. *NeurIPS*.

Graves, A. (2016). Adaptive computation time for recurrent neural networks. *arXiv:1603.08983*.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. *ICML*.

Ha, D., & Schmidhuber, J. (2018). World models. *NeurIPS*.

Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2020). Dream to control: Learning behaviors by latent imagination (Dreamer). *ICLR*.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *NeurIPS Deep Learning Workshop*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. *arXiv:2106.09685 (ICLR 2022)*.

Irving, G., Christiano, P., & Amodei, D. (2018). AI safety via debate. *arXiv:1805.00899*.

Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., & Lewis, M. (2020). Generalization through memorization: Nearest neighbor language models. *ICLR*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. (2017). Overcoming catastrophic forgetting in neural networks (EWC). *PNAS*, 114(13), 3521–3526.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning (UCT). *ECML*.

Kolodner, J. L. (1993). *Case-Based Reasoning*. Morgan Kaufmann.

Kumaran, D., Hassabis, D., & McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20(7), 512–534.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33(1), 1–64.

Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. *ICML*.

Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS*.

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching (experience replay). *Machine Learning*, 8, 293–321.

Madaan, A., Tandon, N., Gupta, P., et al. (2023). Self-Refine: Iterative refinement with self-feedback. *NeurIPS*.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex. *Psychological Review*, 102(3), 419–457.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24, 109–165.

Miall, R. C., Weir, D. J., Wolpert, D. M., & Stein, J. F. (1993). Is the cerebellum a Smith predictor? *Journal of Motor Behavior*, 25(3), 203–216.

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529–533.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 54–71.

Pomerleau, D. A. (1991). Efficient training of artificial neural networks for autonomous navigation (ALVINN). *Neural Computation*, 3(1), 88–97.

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1), 79–87.

Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2), 123–146.

Ross, S., Gordon, G., & Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning (DAgger). *AISTATS*.

Schick, T., Dwivedi-Yu, J., Dessì, R., et al. (2023). Toolformer: Language models can teach themselves to use tools. *NeurIPS*.

Schrittwieser, J., Antonoglou, I., Hubert, T., et al. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model (MuZero). *Nature*, 588, 604–609.

Schuster, T., Fisch, A., Gupta, J., Dehghani, M., Bahri, D., Tran, V. Q., Tay, Y., & Metzler, D. (2022). Confident adaptive language modeling. *NeurIPS*.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ICLR*.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *NeurIPS*.

Silver, D., Huang, A., Maddison, C. J., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529, 484–489.

Stern, M., Shazeer, N., & Uszkoreit, J. (2018). Blockwise parallel decoding for deep autoregressive models. *NeurIPS*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning (options). *Artificial Intelligence*, 112(1–2), 181–211.

Thrun, S. (1998). Lifelong learning algorithms. In *Learning to Learn* (pp. 181–209). Springer.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *NeurIPS*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR*.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., & Anandkumar, A. (2023). Voyager: an open-ended embodied agent with large language models. *arXiv:2305.16291 (TMLR 2024)*.

Wei, J., Wang, X., Schuurmans, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*.

Wolpert, D. M., Miall, R. C., & Kawato, M. (1998). Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9), 338–347.

Wu, P., Escontrela, A., Hafner, D., Abbeel, P., & Goldberg, K. (2023). DayDreamer: World models for physical robot learning. *CoRL 2022 / PMLR*.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., & Press, O. (2024). SWE-agent: agent–computer interfaces enable automated software engineering. *NeurIPS*.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. *ICLR 2023*.