FABIUS

WHITEPAPER · V1.1.0

Fabius

The autonomous AI agent that runs on every major model — fifteen coordinated skills, a research-grounded decision policy, and the mathematics of knowing when to stop.

Ariel Shemesh

fabius-landing.vercel.app · private · provenance-sealed

ABSTRACT

Large language models, used as coding and reasoning agents, fail in two opposite directions: they over-explain and over-engineer, or — when told merely to "be concise" — they cut the validation, security, and accessibility that make an artifact correct. **Fabius** is a single operating stance, realized as an autonomous AI agent — fifteen coordinated skills under one router — that resolves this tension along one axis: *scout wide in what you investigate, strike narrow in what you ship.*

This whitepaper presents fabius as a system. It gives the architecture (a router — the *praetorium* — that dispatches on three axes over an always-on lean core, four engineering specialists, and nine domain verticals), the decision policy (a proven core of eighteen routing rules drawn from the agent-research canon, plus four operational extensions for model-tier dispatch, long-horizon loops, verticals, and corpus externalization), and — the centre of the document — the **mathematics** beneath each core rule: every gate reduced to a formal statement, proved, and adversarially checked. We then prove the core *coheres*: all eighteen rules are one expected-loss / value-of-information threshold applied to disjoint variables, composed in an acyclic pipeline that routes every task and terminates. A strict honesty ledger separates what the source papers *measured* from what fabius *borrows by analogy* — and marks the operational extensions as the working edge, not yet folded into the proof. We close with the **fabius benchmark** — one reproducible test read out on three panels: blind-judged quality on the four newest Claude models, objective deliverable tests that execute generated code against hidden suites and grade domain work against factual checklists, and blind external-model demos across families. Structure

beats brevity, and the advantage grows as the model's default discipline drops — on 20–35% less output.

Fabius is realized as an **autonomous agent**: it runs on every major model (Anthropic, OpenAI, Google, Mistral, Groq), executes the loop — *Scout → Plan → Strike → Prove → Record* — with an independent, execution-grounded verifier, and is operated from the **synapse console**. Three surfaces, one identity: a public face, a private provenance-sealed brain, and the console that runs it. This paper formalizes the decision policy — the router — at the agent's core.

Version 1.1.0 · July 2026 · figures computed (numpy → SVG), reproducible

1 · Introduction

A capable model is not the same as a disciplined one. Hand a frontier LLM a real engineering task and it tends to do too much: it pads the explanation, builds a plugin system for a single flag, hedges every claim. Correct the over-building with the obvious instruction — "*be concise, write minimal code*" — and a second failure appears: the model now cuts the input validation, the error handling, the accessibility, the security. Brevity is not discipline. It trades one defect for another.

Fabius is named for Quintus Fabius Maximus, whose doctrine against Hannibal was to **scout the whole field and fight only the battle that mattered** — never the pitched engagement the stronger enemy wanted. Translated to an agent, the doctrine is a single axis that dissolves the thoroughness-versus-minimalism tension:

Scout wide in what you investigate. Strike narrow in what you ship.

Fan out to understand and to verify; deliver the smallest correct artifact; explain it in the fewest exact words. These never conflict, because they live on different axes — process and memory make you wide, lean makes you narrow. Fabius realizes this as one agent of fifteen coordinated skills: a router, an always-on lean core, four engineering specialists (process, design, agent-engineering, memory), and nine domain verticals (go-to-market, defensive security, game craft, on-chain + sealing, automation, science, AI/ML engineering, markets & finance, cross-model council).

The contribution of this whitepaper is to show that the stance is not a vibe. Section 2 gives the system architecture and the single-owner coordination contract that keeps fifteen skills from contradicting one another. Section 3 states the decision policy: a proven core of eighteen operational rules, each drawn from a specific result in the agent-research literature, and four operational extensions at the working edge. Section 4 — the heart of the document — supplies the **mathematics** under each core rule, every theorem proved and adversarially verified, with a hard line between results that *govern* a routing decision and theorems borrowed only for their *shape*. Section 5 proves the eighteen core rules **compose into one coherent decision system**. Section 6 reports the fabius benchmark — one reproducible test in three panels: blind-judged quality, objective deliverable tests, and cross-family demos. The discipline of the whole is in Section 7: **measured, not claimed**.

2 · The system

Fabius is one agent, not a bundle of plugins. A single router — the *praetorium* — dispatches fourteen coordinated capability layers over a thin supporting spine (fifteen skills in all, counting the router), so the agent gains end-to-end capability under one stance. The router does not just pick a layer; it dispatches on three axes at once: **which layer(s)**, **how much machinery** (the capability ladder of §4.1), and **which model tier**.

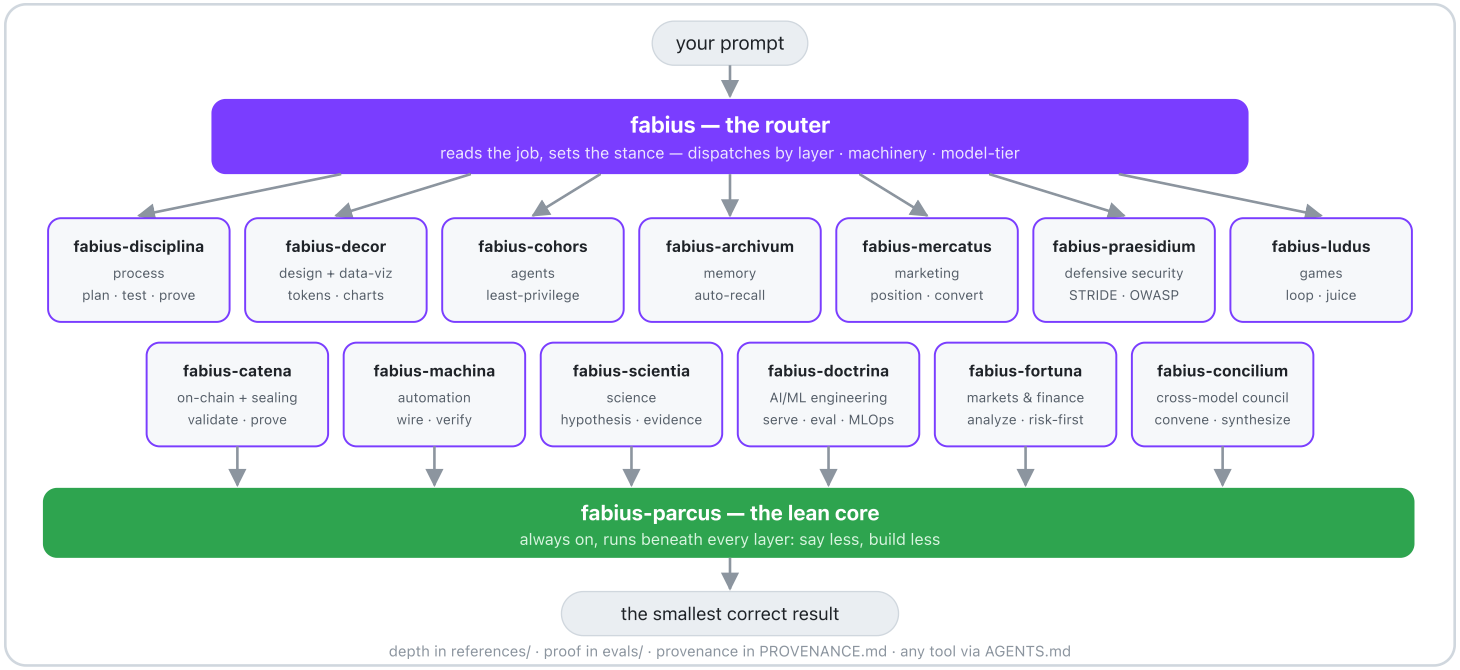

Figure 0 — the layer model. Your prompt enters the `fabius` router (the *praetorium*), which dispatches by layer, machinery, and model-tier to the specialist a task needs — four engineering specialists (`disciplina` process, `decor` design + data-viz, `cohors` agents, `archivum` memory) and nine domain verticals (`mercatus` go-to-market, `praesidium` defensive security, `ludus` games, `catena` on-chain + sealing, `machina` automation, `scientia` science, `doctrina` AI/ML engineering, `fortuna` markets + finance, `concilium` cross-model council) — all running on the always-on `fabius-parcus` lean core, producing the smallest correct result. Depth loads on demand from `references/` (indexed by `CORPUS.md`); the benchmark is in `evals/`; the stance ports to any tool via `AGENTS.md`.

2.1 · Fifteen skills, one stance

Each skill is a thin operating contract; its depth lives under `references/`, indexed by `CORPUS.md`, and loads only on demand.

Skill	Role	What it delivers
`fabius`	router · praetorium	reads the job, sets the stance, dispatches by layer + machinery + model-tier
`fabius-parcus`	lean core, always on	terse output · the YAGNI ladder · surgical changes · the never-trim floor
`fabius-disciplina`	engineering process	brainstorm → plan → test-first → prove · root-cause debugging
`fabius-decor`	design + data-viz	one-accent laws · token vocabulary · mobile-first · data-ink charts (figura)
`fabius-cohors`	agent engineering	definition schema · least privilege · five orchestration patterns
`fabius-archivum`	persistent memory	per-project memory · the LLM-wiki · index + log retrieval · recall

Skill	Role	What it delivers
`fabius-mercatus`	go-to-market	positioning · message-to-awareness match · proof over adjectives · a one-action funnel
`fabius-praesidium`	defensive security	STRIDE per boundary · the OWASP pass · severity → fix → proof findings
`fabius-ludus`	game craft	the core loop first · deliberate juice · state as a machine · jam-sized scope
`fabius-catena`	on-chain + sealing	account-validation-first contracts (EVM + Solana) · money-safe transactions · verifiable provenance sealing
`fabius-machina`	automation	deterministic workflow glue · discover-from-live-schema → build → validate AND verify → activate
`fabius-scientia`	science	competing falsifiable hypotheses · source-grounded database lookups · reproducible field-standard pipelines
`fabius-doctrina`	AI/ML engineering	the model lifecycle — train/fine-tune → evaluate → serve → monitor · held-out eval + blind judges · vLLM-class serving · MLOps
`fabius-fortuna`	markets & finance	risk-first analysis · fundamental + technical + quantitative · valuation · honest out-of-sample backtesting (analysis, not advice)
`fabius-concilium`	cross-model council	convene N models on one question · first opinions → anonymized peer-review → chairman synthesis · ensemble epistemics, gated on cost (N+N+1 calls)

Latin, for the curious: **parcus**, frugal · **disciplina**, training · **decor**, what is fitting · **cohors**, the cohort · **archivum**, the record office · **mercatus**, the marketplace · **praesidium**, the garrison · **ludus**, the game and the school where you drill it · **catena**, the chain · **machina**, the working mechanism · **scientia**, knowledge won by method · **doctrina**, the body of learning (a machine taught to learn) · **fortuna**, fortune and the turn of the market · **concilium**, the summoned council.

2.2 · Single owner, zero overlap

The coordination contract is one rule: **each rule has exactly one owning layer; every other layer links to it instead of restating it.** Planning, test discipline, and the clarifying-question procedure are owned by `disciplina`; lean prose, the YAGNI ladder, and the never-trim guardrail list by `parcus`; routing and the kill-switch by `fabius`. The line between core and vertical is drawn the same way: `parcus` owns the *never-trim security floor* (don't cut validation), while `praesidium` owns the *active* security work (model the threat, name the check, prove it closed) and references that floor rather than restating it. `parcus` is the only always-on layer: it composes *underneath* whatever task layer the router selects and never competes for a task verb. This single-owner discipline is what makes the formal coherence of Section 5 possible — no two layers write the same rule with opposite intent.

2.3 · The corpus — the brain holds the index, not the library

The depth behind the skills — a 69-brand design teardown library, a 200-plus-agent production catalog with a vector index, a knowledge engine, and the marketing, security, and game-craft playbooks — is large. Fabius keeps the *install* lean by separating brain from library: a single fabius-branded index (`CORPUS.md`) sits over every capability library, and the router holds only the index. On a task it reads the index, pages in the one matching slice, and never loads the bulk (the memory discipline of §4.5, rules R9 · M7 · M9). Honest current state: three libraries still ship bundled under `references/` today and are migrating behind the index (§4.7, M9) — the contract is built so they can

externalize without any skill changing how it reaches them. The standing rule is that adding a capability is adding a row to the index, not a megabyte to the install — so the library scales without bloating the brain.

3 · The decision policy

A stance is only as good as the decisions it makes: *which* layer to route to, *when* to spend a tool call or a second agent, *how long* to keep refining, *what* to load from memory. Fabius's router carries an explicit policy of eighteen rules — ten routing rules (R1–R10) and eight orchestration / memory rules (M1–M8) — each drawn from a specific result in the agent-research canon and tagged for honesty.

3.1 · Route by classification, then climb one rung

Specialist selection is reproducible, not vibes. Fabius opens every non-trivial task by naming its load on three axes — **Memory, Tools/Action, Planning** — the universal capability axes from Wang et al.'s survey of autonomous agents **(R1)**. Memory routes to `archivum`, Tools to `cohors`, Planning to `disciplina`; zero axes loaded stays in the lean `parcus` core. A task in a named vertical also routes on a **Domain** axis to its owner — UI/data-viz to `decor`, go-to-market to `mercatus`, defensive security to `praesidium`, games to `ludus`, on-chain/provenance to `catena`, automation to `machina`, science to `scientia` — where *domain picks what* while the three load axes pick *how*, and the vertical runs as a studio (R13). The classification *is* the routing rationale.

Then it climbs a capability ladder one rung at a time — `inline → one tool → retrieval → plan → single subagent → swarm` — adding the smallest rung the classification demands and stopping **(R2)**. The efficiency surveys describe a cost–capability frontier with steep diminishing returns; fabius operates at the *knee*, not the tail. The formal version of "stop at the knee" is proved in §4.1.

3.2 · The eighteen rules at a glance

Rule	The decision it makes	Source paper	grounding
R1	classify a task on {Memory, Tools, Planning}, route each loaded axis	Wang survey	direct
R2	climb the capability ladder one rung; stop at the knee	efficiency surveys	analogy
R3	call a tool only when you can name the wrong answer it prevents	Toolformer	analogy
R4	on an ambiguous task, keep 2–3 routes alive; collapse as signal sharpens	Flow Matching	analogy
R5	reason → act → observe; never act on an assumed result	ReAct	direct
R6	plan in placeholders; bind tool calls only after the plan is fixed	Chain of Abstraction	analogy
R7	branch only when a cheap evaluator can rank unfinished candidates	Tree of Thoughts	direct
R8	refine only on a verifiable signal; signal type sets the budget	Reflexion · Self-Refine	direct
R9	retrieve on demand; read the index, page in only the matching slice	MemGPT	direct
R10	state breadth tasks once; hard-steer only narrow contracts	Classifier-Free Guidance	analogy
M1	spawn a second agent only to prevent a named error	Toolformer-spirit	analogy
M2	add a reducer only when partial results merge (associativity)	Graph of Thoughts	analogy
M3	scale verification depth to measured per-route failure	Consistency Models	analogy
M4	reflect-then-retry; escalate when hypotheses run out	Reflexion	direct
M5	accept a prompt rewrite only on a held-out metric delta	DSPy	direct
M6	promote verified solutions to a reusable skill; query before planning	Voyager	direct

Rule	The decision it makes	Source paper	grounding
M7	page memory explicitly; write only decision-changing facts	MemGPT	direct
M8	rank index ties by relevance, then recency and load-bearingness	Generative Agents	analogy

The reading list behind each rule is the agent-research canon — ReAct, Toolformer, Tree and Graph of Thoughts, Reflexion, Self-Refine, MemGPT, DSPy, Voyager, the Wang and efficiency surveys, and (for the analogies) the sampling mathematics of generative models. Section 4 turns each rule into a theorem.

3.3 · The operational edge — four extensions beyond the proven core

The eighteen rules above are the **proven core**: Section 5's coherence theorem is established over them and them alone. As the system grew to dispatch model tiers, run long-horizon loops, compose verticals, and package its own corpus, it added four **operational extensions**. These are honestly outside the proof — borrowed-by-analogy or system-internal, consistent with the core in practice but not yet formalized or folded into the coherence theorem. They are the working edge.

Rule	The decision it makes	Source
R11	spend the cheapest model tier that holds; escalate on a *verifiable* miss, not a guess — re-tier per sub-task, not per session	efficiency survey · FrugalGPT cascades
R12	long-horizon work runs `step → verify` on a loop with a dual exit gate (completion *and* done-signal) and a hard cycle cap — autonomous, never infinite	Ralph loop · Reflexion stop-condition
R13	a vertical runs a studio — the domain skill leads the WHAT, process plans, execution follows, lean underneath	fabius's own composition rule
M9	externalize the corpus — the brain holds the index, not the library; adding a capability is an index row, not a megabyte	MemGPT paging · memory survey

R11–R13 and M9 are stated in full in §4.7, after the core proofs — with the boundary kept explicit: they are sound and useful, but they have not earned a place inside the theorem yet, and this document does not pretend otherwise.

4 · The mathematics of the policy

Each rule below is reduced to a formal statement and proved. Every proof was written and then **adversarially verified by an independent reviewer** whose only job was to find an error — a process that, in development, caught a sign error in the value-of-information gate, a missing completeness hypothesis in the contraction argument, an over-stated submodular-knapsack bound, and an attribution slip on the ranking principle, all corrected before publication.

Two tags run through the section. **Real-math** means the equation genuinely *governs* the routing decision — decision theory, information theory, optimization, scheduling, algebra. **Analogy** means a correct theorem from another field (generative-model sampling, transport, IR kernels) borrowed for its *shape*; the source paper proves nothing about agents, so fabius takes the shape, not a measured result. One rule (R5) is **qualitative** — a control-flow invariant with no objective function. The honesty boundary is stated inside every analogy proof.

4.1 · Routing, and the value of spending

R1 Routing is a measurable partition of task-space

REAL-MATH

Statement. Let each task t carry a load vector $\ell(t) = (\ell_M, \ell_T, \ell_P) \in \{0,1\}^3$, where ℓ_M, ℓ_T, ℓ_P indicate nonzero load on the Memory, Tools/Action, and Planning axes respectively. Equip the finite label space $L = \{0,1\}^3$ with its discrete σ-algebra 2^L. Then (i) the level sets $C_b = \ell^{-1}(\{b\})$ for $b \in L$ form a measurable partition of task-space into 8 cells; and (ii) under a fixed priority order $P \succ T \succ M$ on loaded axes (with the empty load reserved for parcus), the assignment $\rho : L \to \{\textbf{archivum}, \textbf{cohors}, \textbf{disciplina}, \textbf{parcus}\}$ is total and single-valued.

Proof. (i) The family $\{C_b\}_{b \in L}$ is the collection of preimages of singletons under ℓ. Distinct singletons are disjoint, so $C_b \cap C_{b'} = \ell^{-1}(\{b\} \cap \{b'\}) = \varnothing$ for $b \neq b'$ (mutual exclusivity), and $\bigcup_{b \in L} C_b = \ell^{-1}(L) =$ the whole space (exhaustivity), since ℓ is defined on every task. Each $\{b\} \in 2^L$, so each C_b is measurable; hence $\{C_b\}$ is a measurable partition. As $|L| = 2^3 = 8$, there are exactly 8 cells. (ii) Define $\rho(0,0,0) = \textbf{parcus}$; for $b \neq 000$, let $\rho(b)$ be the layer of the highest-priority loaded axis under $P \succ T \succ M$: planning→disciplina, else tools→cohors, else memory→archivum. Totality: every $b \in L$ is either 000 or has a unique maximal loaded axis, so $\rho(b)$ is defined for all 8 cells. Single-valuedness: the priority order is a strict total order on the three axes, so "highest-priority loaded axis" is unique, giving exactly one image; ρ is a function, not a relation. □

In fabius. R1 computes the 3-bit load of an incoming task and lands it in exactly one of the 8 cells; the empty cell routes to parcus and any loaded cell routes to its dominant specialist under $P \succ T \succ M$, so every task is dispatched to one and only one layer with no fall-through and no contention.

R2 The capability ladder stops at the concavity knee

REAL-MATH

Statement. Let the ladder be $n \in \{0, 1, \ldots, N\}$. Assume value V is discrete-concave, i.e. its forward difference $\Delta V(n) := V(n+1) - V(n)$ is non-increasing in n on $\{0, \ldots, N-1\}$; assume cost C is convex, i.e. $\Delta C(n) := C(n+1) - C(n)$ is non-decreasing; let utility $U := V - C$. Define $n^{\backslash *} = \min\{n : \Delta V(n) \le \Delta C(n)\}$, with $n^{\backslash *} = N$ when that set is empty. Then U attains its maximum at $n^{\backslash *}$; the $\le$ makes a break-even rung not worth adding.

Proof. The marginal utility is $\Delta U(n) = U(n+1) - U(n) = \Delta V(n) - \Delta C(n)$. Since ΔV is non-increasing and ΔC is non-decreasing, $-\Delta C$ is non-increasing, so ΔU is a sum of two non-increasing sequences and is itself non-increasing: U is discrete-concave. For any m,

$$U(m) - U(0) = \sum_{k=0}^{m-1} \Delta U(k).$$

By definition of $n^{\backslash *}$, $\Delta U(k) = \Delta V(k) - \Delta C(k) > 0$ for all $k < n^{\backslash *}$ (those rungs strictly increase U), while for $k \ge n^{\backslash *}$ monotonicity of ΔU and $\Delta U(n^{\backslash *}) \le 0$ give $\Delta U(k) \le 0$ (those rungs do not increase U). Hence partial sums of ΔU rise up to index $n^{\backslash *}$ and never strictly exceed $U(n^{\backslash *})$ afterward: for $m < n^{\backslash *}$, $U(n^{\backslash *}) - U(m) = \sum_{k=m}^{n^{\backslash *}-1} \Delta U(k) > 0$; for $m > n^{\backslash *}$, $U(m) - U(n^{\backslash *}) = \sum_{k=n^{\backslash *}}^{m-1} \Delta U(k) \le 0$. Thus $U(n^{\backslash *}) \ge U(m)$ for all m, so $n^{\backslash *}$ is a global maximizer. Because the defining inequality is $\le$, a rung with $\Delta V(n^{\backslash *}) = \Delta C(n^{\backslash *})$ yields $\Delta U(n^{\backslash *}) = 0$: climbing it cannot raise U, so the smaller rung $n^{\backslash *}$ is selected. □

In fabius. R2 climbs the capability ladder (inline → tool → ReAct → ToT/GoT → multi-agent) only while the marginal value of the next rung strictly exceeds its marginal cost, halting at the smallest sufficient rung $n^{\backslash *}$; the $\le$ tie-break keeps the agent inline (or one rung lower) whenever a heavier rung merely breaks even.

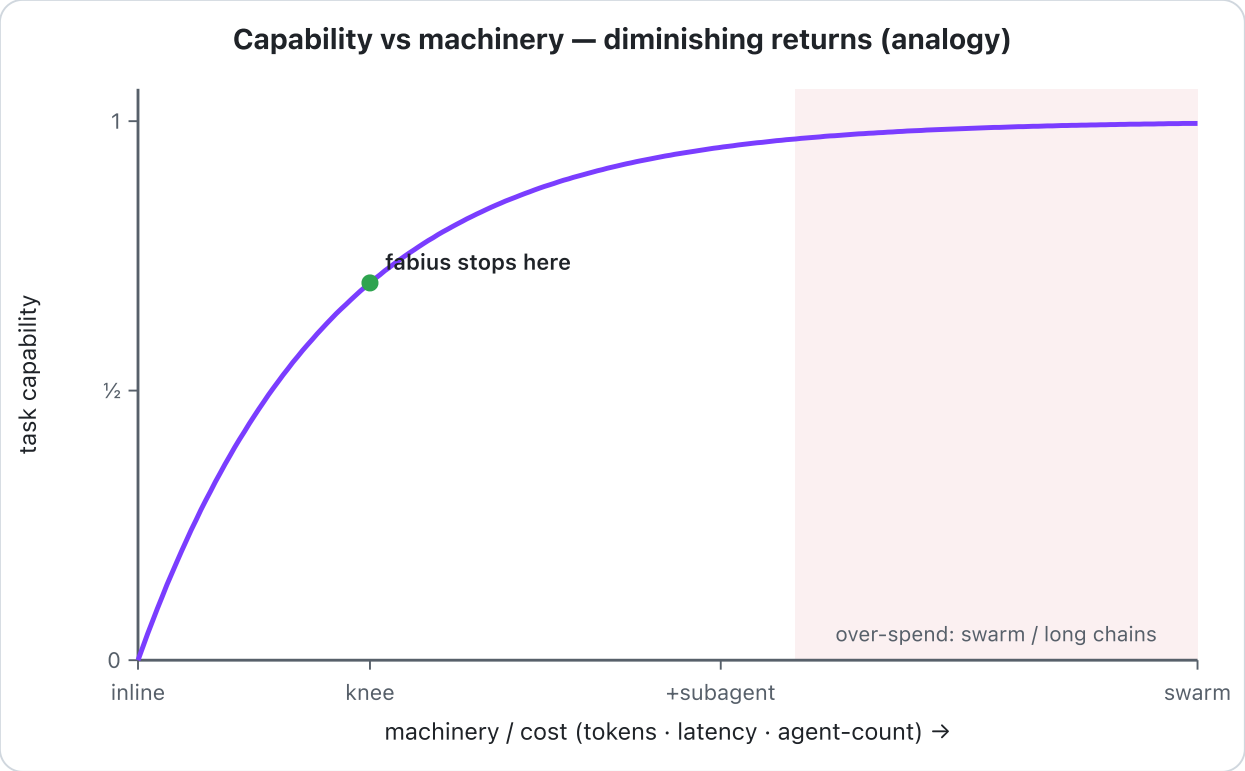

Figure 1 — analogy. Capability scales sub-linearly with machinery; fabius adds the smallest sufficient rung and targets the *knee* (R2), never the tail. The diminishing-returns shape is asserted directionally by the efficiency surveys — this exact curve is not fitted to fabius.

R3 A call must clear its expected value of information

REAL-MATH

Statement. Fix a loss $L \geq 0$ on outcomes and let an agent face two mutually exclusive actions. *Inline*: answer now, incurring expected loss $a := \mathbb{E}[L \mid \textbf{inline}]$. *Call*: pay a fixed cost $c_{\text{call}} > 0$ for an observation, then act optimally on it, incurring total expected loss $b + c_{\text{call}}$ where $b := \mathbb{E}[L \mid \textbf{call}]$ is the post-observation expected loss under the call's own optimal policy. Both expectations are taken under the agent's prior; c_{call} is independent of the realized loss. **Claim.** The risk-minimizing action is the call iff the net value of information strictly exceeds its cost: $a - b > c_{\text{call}}$.

Proof. The optimum is $\min\{a,\ b + c_{\text{call}}\}$. The call is strictly preferred iff $b + c_{\text{call}} < a$, i.e. $a - b > c_{\text{call}}$; ties ($=$) and $a - b < c_{\text{call}}$ keep inline. Define $\textbf{EVOI} := a - b$, the gross expected loss reduction the call buys. The gate routes to the call exactly when $\textbf{EVOI} > c_{\text{call}}$, the standard cost-of-information comparison (Howard's value-of-information; Raiffa–Schlaifer). □

Why EVOI here is two-sided, unlike EVPI/EVSI ≥ 0. Both classical EVPI and classical EVSI are nonnegative, and for the same reason: the decision-maker is assumed to *retain the prior-optimal (inline) action* and adopt the sample's recommendation only when it improves on it. Formally, with perfect information the agent picks per realization ω over an action set $\mathcal{A}$ that *still contains* inline, so $\min_{\alpha \in \mathcal{A}} L(\alpha, \omega) \leq L(\textbf{inline}, \omega)$; taking expectations, $\mathbb{E}[\min_\alpha L] \leq a$ and $\textbf{EVPI} = a - \mathbb{E}[\min_\alpha L] \geq 0$. Sample information gives $0 \leq \textbf{EVSI} \leq \textbf{EVPI}$ by the same retain-the-default argument plus the data-processing inequality. The R3 quantity $a - b$ is *not* of this protected form: b is the loss under the call's *own* optimal policy, whose action set need not contain the inline answer and whose noisy observation may mislead. Hence $a - b$ can be *negative* — a faulty tool or wrong retrieval gives $b > a$, so $\textbf{EVOI} = a - b < 0 < c_{\text{call}}$ and the gate fails *even at zero cost*. The inequality is therefore binding in both directions: a costless-but-faulty call is rejected, and a sound-but-expensive call is rejected unless its information clears c_{call}. □

In fabius. R3 fires a tool/skill call only when its estimated net value $a - b$ exceeds the call's cost c_{call} (latency + tokens + failure risk); a faulty or redundant call with $b \geq a$ is rejected even when free, which is why this gate is two-sided rather than the trivially-nonnegative EVPI/EVSI.

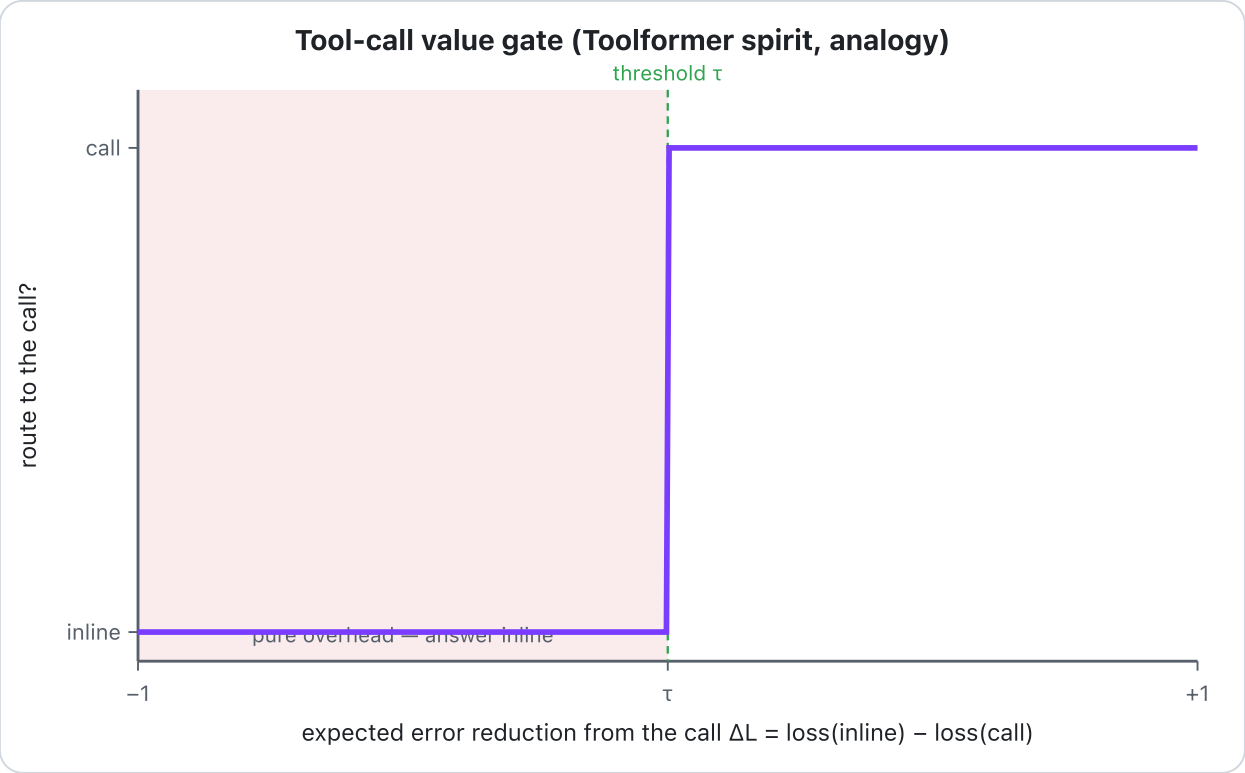

Figure 2 — analogy. The value-of-information gate (R3): route to the call only when expected error-reduction clears its cost; below threshold is pure overhead. Toolformer computes its loss-reduction filter at training time, which fabius lacks at routing time — so it asks instead which wrong answer the call prevents.

M1 A second agent needs decorrelated error to pay

REAL-MATH

Statement. Let a task be completed by one agent (the author) at expected loss $E[L \mid \text{single}]$. Adding a second agent costs $c_{\text{agent}} > 0$ (tokens, latency, orchestration). Under the R3 marginal-value gate, spawn the second agent iff

$$E[L \mid \text{single}] - E[L \mid +\text{agent}] \;>\; c_{\text{agent}}.$$

Specialize to a reviewer that only catches errors. Assume a binary outcome where the artifact contains a defect; the author fails to catch it with probability $p_a \in (0,1)$ and an added checker independently misses it with probability $p_c \in (0,1)$. Loss $= 1$ on an uncaught defect, 0 otherwise. Then under independence the joint-miss probability is $p_a p_c$, giving strict gain $p_a(1-p_c)$; under error-correlation ρ the gain erodes monotonically in ρ. The load-bearing assumption is independence — i.e. a reviewer who did not write the artifact.

Proof. With one agent, $E[L \mid \text{single}] = p_a$. The defect survives only if both the author and the checker miss it. Let $A = \{\text{author misses}\}$, $C = \{\text{checker misses}\}$, with $\Pr(A) = p_a$, $\Pr(C) = p_c$. Independence gives

$$E[L \mid +\text{agent}] = \Pr(A \cap C) = \Pr(A)\Pr(C) = p_a p_c.$$

Hence the marginal benefit is

$$\Delta = p_a - p_a p_c = p_a(1-p_c) > 0 \quad \text{for } p_c < 1,$$

and $p_a p_c < p_a$ strictly. The R3 gate fires iff $p_a(1-p_c) > c_{\text{agent}}$. Now drop independence. By Fréchet/covariance decomposition,

$$\Pr(A \cap C) = \Pr(A)\Pr(C) + \operatorname{Cov}(\mathbf{1}_A, \mathbf{1}_C) = p_a p_c + \rho\sqrt{p_a(1-p_a)p_c(1-p_c)},$$

where $\rho = \operatorname{Corr}(\mathbf{1}_A, \mathbf{1}_C)$. Then

$$\Delta(\rho) = p_a - \Pr(A \cap C) = p_a(1-p_c) - \rho\sqrt{p_a(1-p_a)p_c(1-p_c)},$$

so $\partial\Delta/\partial\rho = -\sqrt{p_a(1-p_a)p_c(1-p_c)} \le 0$: the gain decreases monotonically as the two agents' errors become more correlated. At the comonotone extreme $\rho \to +\sqrt{\frac{p_a(1-p_c)}{p_c(1-p_a)}}$ (with $p_c \ge p_a$) we get $\Pr(A \cap C) \to p_a$ and $\Delta \to 0$: a perfectly correlated checker — e.g. the same agent re-reading its own work — adds nothing while still costing c_{agent}. □

In fabius. cohors spawns a second agent only when the predicted defect-catch gain clears c_{agent}, and it requires the reviewer to be a fresh agent that did not author the artifact — because correlated error (the author re-checking itself) drives $\Delta \to 0$ and fails the R3 gate.

R7 Branch only when sample information beats its cost

REAL-MATH

Statement. Let x be an unknown state with prior distribution $P(x)$, let a range over a finite action set A, and let $U(a, x)$ be a utility with $\mathbb{E}_x[|U(a, x)|] < \infty$ for every a. Define the no-information value $V_0 = \max_a \mathbb{E}_x[U(a, x)]$. A signal s (with any finite branching factor, i.e. any number of possible values) has joint law $P(s, x)$; define the sample-information value $V_1 = \mathbb{E}_s\big[\max_a \mathbb{E}_{x|s}[U(a, x)]\big]$ and $\mathbf{EVSI} = V_1 - V_0$. Let $\mathbf{EVPI} = \mathbb{E}_x[\max_a U(a, x)] - \max_a \mathbb{E}_x[U(a, x)]$. Then $\mathbf{0 \le EVSI \le EVPI}$, and $\mathbf{EVSI = 0}$ whenever $s \perp x$, for any branching factor.

Proof. *Lower bound.* Fix any $a^\star \in \arg\max_a \mathbb{E}_x[U(a, x)]$. For every realized s, $\max_a \mathbb{E}_{x|s}[U(a, x)] \ge \mathbb{E}_{x|s}[U(a^\star, x)]$. Taking $\mathbb{E}_s$ and using the tower property $\mathbb{E}_s\mathbb{E}_{x|s}[U(a^\star, x)] = \mathbb{E}_x[U(a^\star, x)] = V_0$, we get $V_1 \ge V_0$, so $\mathbf{EVSI \ge 0}$. *Upper bound.* For each s, $\max_a \mathbb{E}_{x|s}[U(a, x)] \le \mathbb{E}_{x|s}[\max_a U(a, x)]$, since the max of expectations is at most the expectation of the pointwise max (Jensen for the convex max functional). Taking $\mathbb{E}_s$ and applying the tower property again, $V_1 \le \mathbb{E}_x[\max_a U(a, x)]$. Subtracting V_0 gives $\mathbf{EVSI \le EVPI}$. *Independence.* If $s \perp x$, then $P(x \mid s) = P(x)$, so $\mathbb{E}_{x|s}[U(a, x)] = \mathbb{E}_x[U(a, x)]$ for every a and every s; hence $\max_a \mathbb{E}_{x|s}[U(a, x)] = V_0$ is constant in s, giving $V_1 = V_0$ and $\mathbf{EVSI = 0}$ regardless of how many values s takes. □

In fabius. R7 spawns a branch (Tree/Graph-of-Thoughts fan-out) only when an evaluator's expected information value exceeds its search cost — $\mathbf{EVSI} > c_{\text{search}}$; since $\mathbf{EVSI = 0}$ for an evaluator uninformative about the answer (no matter how wide the branching), fabius collapses to a single ReAct chain when no cheap, discriminating evaluator exists.

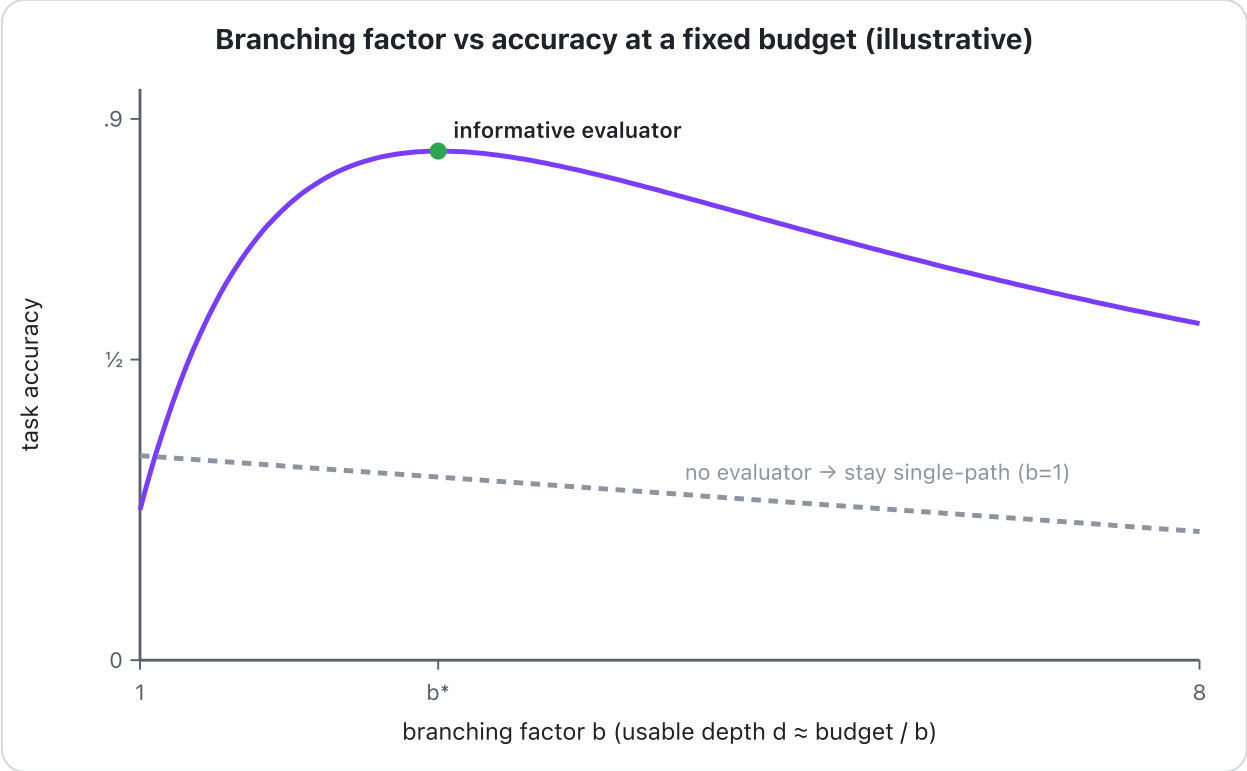

Figure 3 — illustrative. With an informative evaluator, accuracy peaks at an interior branching factor (R7); with none, more branches only cost depth. Tree of Thoughts demonstrates the evaluator-gated benefit empirically but does not publish this curve.

4.2 · Execution and scheduling

R5 Reason→act→observe is an invariant, not an optimum

QUALITATIVE

Statement. Model an agent run as a finite trace $\tau = e_1 e_2 \cdots e_n$ of events, each a Reason, Action, or Observation. Let $\mathbf{prov}(a)$ be the set of facts an Action a reads as preconditions, and let $\mathbf{real}(\tau, k)$ be the facts established by Observations strictly before position k. Define the grounding predicate G: a trace satisfies G iff (i) for every Action $e_k = a$, $\mathbf{prov}(a) \subseteq \mathbf{real}(\tau, k)$ and $\mathbf{prov}(a)$ contains no fact produced only by a Reason or by an as-yet-unobserved Action (no action is stacked on an assumed result), and (ii) progress: after any window of ≈ 3 consecutive cycles with no change to $\mathbf{real}$ (no-movement), the next event is a Reason that re-plans. Claim: G is a conjoined *safety + liveness* invariant, and there exists no scalar objective $J(\tau)$ that G maximizes.

Proof. This is an invariant, not an optimization: no quantity is maximized. Safety component (i) is a prefix-closed predicate — if it holds on τ it holds on every prefix, since the gating condition at position k refers only to events before k; thus violation is detected at the first ungrounded Action and "nothing bad" (acting on a fact no Observation ever returned) is excluded as a per-step guard. By the standard Alpern–Schneider decomposition, (i) is a pure safety property: its bad set is a finite-prefix property. Component (ii) is liveness: every infinite admissible continuation must eventually contain a re-plan, ruling out the unbounded stall in which the agent loops on stale assumptions — it guarantees "something good eventually happens." Their conjunction is the ReAct discipline. Now suppose, for contradiction, an objective J existed with $G(\tau) \iff \tau \in \arg\max J$. Two grounded traces solving the same task by different tool orders both satisfy G yet are incomparable under any single J unless J is constant on the feasible set; and a constant J is maximized equally by ungrounded traces, contradicting G's rejection of them. Hence G is definable only as a constraint on admissible traces, not as the argmax of any reward. □

In fabius. R5 gates every tool call in disciplina/cohors on the most recent real Observation — never on a Reason-asserted or pending result — and forces a re-plan after roughly three no-movement cycles; it is enforced as an admissibility check on the control flow, never traded off against a score.

In fabius. Because it maximizes nothing, R5 is the one qualitative rule among the eighteen: the other seventeen route by comparing a quantity (cost, value, gain, EVPI), whereas R5 only ever answers admissible/inadmissible.

R6 Plan-then-bind collapses the critical path

REAL-MATH

Statement. Let an agent execute n independent tool calls, where call i incurs reasoning cost $r_i \geq 0$ (to formulate or bind the call) and tool latency $t_i \geq 0$ (to obtain the result). Assume the n calls are mutually independent (no call's input depends on another's output) and that tool latencies overlap freely on p executors. *Bind-as-you-go* interleaves reason-then-await per call, forcing a total chain $T_{\text{serial}} = \sum_{i=1}^{n}(r_i + t_i)$. *Plan-then-bind* first emits all reasoning against abstract placeholders, $\sum_{i=1}^{n} r_i$, then dispatches the now-independent calls concurrently. With $p \geq n$ executors, $T_{\text{plan}} = \sum_{i=1}^{n} r_i + \max_i t_i$. For n identical calls $(r_i = r,\ t_i = t)$, the speedup is $S = T_{\text{serial}}/T_{\text{plan}} = \dfrac{n(r+t)}{nr+t}$, and $S \to n$ as $r/t \to 0$. General lower bound (Brent/Graham): $T_{\parallel} \geq \max\left(\text{span},\ \text{work}/p\right)$.

Proof. Model the computation as a DAG: each call i is a chain $\rho_i \to \tau_i$ of weights r_i then t_i; independence means no edges cross between chains. *Serial.* A single executor that binds-as-you-go imposes a total order over all $2n$ nodes, so its makespan equals their summed weight $\sum_i (r_i + t_i) = T_{\text{serial}}$. *Plan.* Reasoning emitted by one agent is itself serial, contributing $\sum_i r_i$. After it completes, the τ_i form an antichain (pairwise incomparable); with $p \geq n$ each runs on its own executor starting simultaneously, so they finish at $\max_i t_i$. Hence $T_{\text{plan}} = \sum_i r_i + \max_i t_i$. *Speedup.* For the equal case, $S = \frac{n(r+t)}{nr+t}$; fixing t and letting $r \to 0$, $S \to \frac{nt}{t} = n$; equivalently $S = n \cdot \frac{r+t}{nr+t} \uparrow n$ monotonically as $r/t \downarrow 0$. *Lower bound.* The span (longest weighted path) here is $\sum_i r_i + \max_i t_i$; no schedule beats its own critical path, giving $T_{\parallel} \geq \text{span}$. The work (total weight) is $\sum_i r_i + \sum_i t_i$; p executors clear it no faster than work/p, giving $T_{\parallel} \geq \text{work}/p$. Taking the max yields the Brent/Graham bound, which T_{plan} attains at $p \geq n$. □

In fabius. R6 routes any task with two or more mutually independent tool/sub-agent calls to plan-then-bind: emit all reasoning against placeholders first, then fan the calls out concurrently — paying $\sum r_i + \max t_i$ instead of $\sum (r_i + t_i)$.

Honesty boundary. The scheduling law — $T_{\text{plan}} = \sum r_i + \max_i t_i$ and $T_{\parallel} \geq \max(\text{span}, \text{work}/p)$ — is exact and proven here over the DAG. But the speedup *magnitude* $S \to n$ was demonstrated empirically for single-tool Chain-of-Abstraction; transferring that figure to fabius's multi-agent fan-out is an analogy on the curve. We claim the law, not the number: independence and free overlap may fail (shared rate limits, $p < n$, latency variance), and no fabius measurement of the realized S exists.

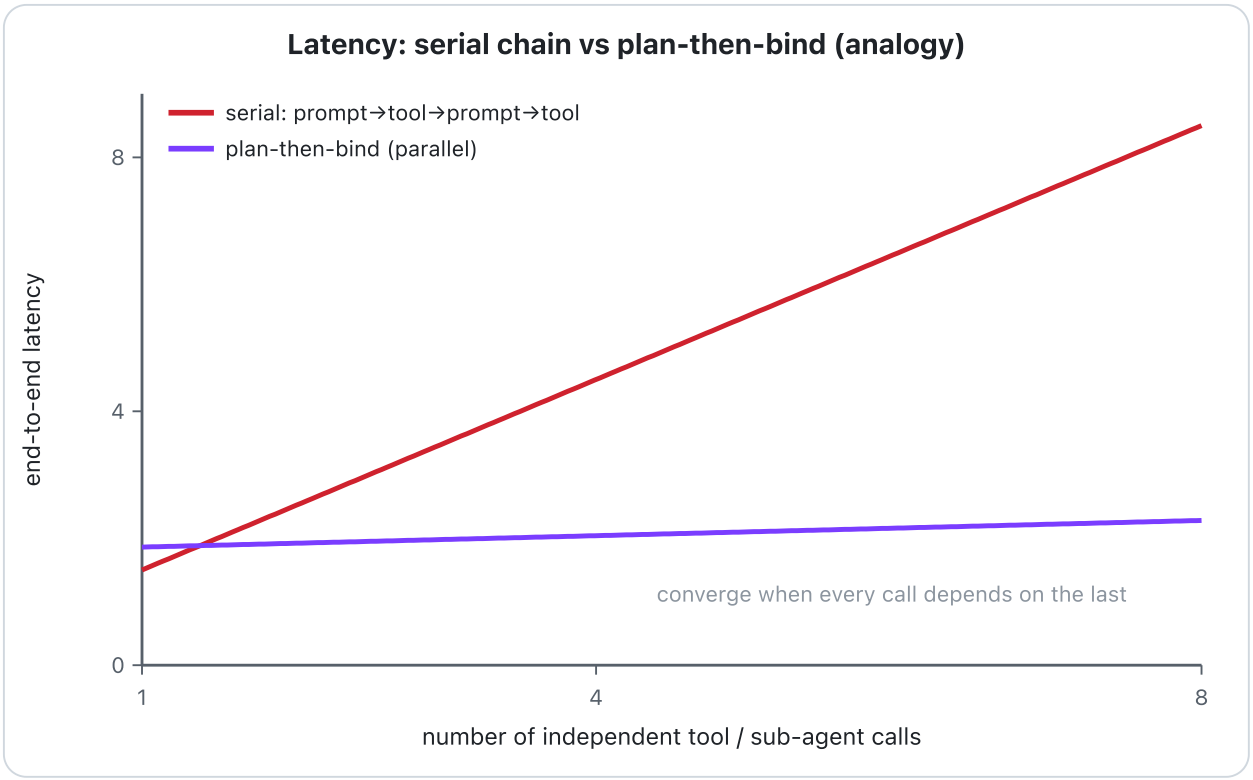

Figure 4 — analogy. Bind-as-you-go grows latency linearly with the number of tool calls; plan-then-bind (R6) stays near-flat by overlapping independent calls. The flat curve is Chain-of-Abstraction's prediction, unmeasured in fabius's multi-agent fan-out; the lines converge when every call depends on the last.

M2 A reducer is licensed exactly by associativity

REAL-MATH

Statement. Let S be a set with a binary combiner $\oplus : S \times S \to S$, and let inputs be finite nonempty sequences over S. For a sequence $x = (x_1, \ldots, x_n)$ define the serial left-fold $F(x) = (((x_1 \oplus x_2) \oplus x_3) \cdots \oplus x_n)$, and let $R(x)$ be any order-preserving parallel reduction: the value at the root of an arbitrary binary tree whose left-to-right leaf order is x, with each internal node evaluating $\oplus$ on its children. Then $R(x) = F(x)$ for all inputs and all trees $\iff$ $\oplus$ is associative, i.e. $(S, \oplus)$ is a semigroup. When $\oplus$ is associative, R is well-defined (split-point-independent) and satisfies the concatenation law $R(A \| B) = R(A) \oplus R(B)$ for all nonempty A, B; thus R is a semigroup homomorphism. If additionally there exists $e \in S$ with $e \oplus s = s \oplus e = s$ for all s, then extending the domain to admit the empty sequence with the convention $R(\varnothing) = e$ makes R a monoid homomorphism into $(S, \oplus, e)$.

Proof. ($\Leftarrow$) Suppose associativity. We show every tree equals F by induction on n. For $n \le 2$ all order-preserving trees are identical, so $R = F$. For $n \ge 3$, any tree splits as $\oplus(R(x_{1:k}), R(x_{k+1:n}))$ for some $1 \le k < n$; by the inductive hypothesis each subtree equals its left-fold, so $R(x) = F(x_{1:k}) \oplus F(x_{k+1:n})$. The generalized associative law (provable by a second induction: full reparenthesization invariance follows from the single rebracketing $(a \oplus b) \oplus c = a \oplus (b \oplus c)$) gives $F(x_{1:k}) \oplus F(x_{k+1:n}) = F(x)$. Hence $R = F$ for all trees, so R is split-point-independent. The concatenation law is exactly the case where the split point is the A / B boundary: $R(A \| B) = F(A \| B) = F(A) \oplus F(B) = R(A) \oplus R(B)$ for nonempty A, B — a semigroup homomorphism requiring no identity. If a monoid identity e exists, setting $R(\varnothing) = e$ preserves $R(A \| B) = R(A) \oplus R(B)$ when either side is empty (since e is two-sided), upgrading R to a monoid homomorphism; the empty sequence is otherwise excluded from the domain and e plays no role in the nonempty law. ($\Rightarrow$) Contrapositive: if $\oplus$ is non-associative, pick a, b, c with $(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$. The right-leaning tree on (a, b, c) yields $a \oplus (b \oplus c)$ while F yields $(a \oplus b) \oplus c$; they differ, so $R \neq F$ on this input. Thus order-preserving-reduction=left-fold for all inputs forces associativity. □

In fabius. M2 dispatches a single reducer/synthesis agent over parallel branches only when the combiner is associative (results genuinely merge, e.g. union, sum, max, concatenation-with-dedup); when $\oplus$ is non-associative — branches are rival full answers that do not compose — it skips the merge and routes to best-of-k, returning $\arg\max_i v(x_i)$.

Honesty boundary. The associativity theorem above is proven about algebra and licenses the merge-vs-best-of-k decision exactly. What does NOT transfer as proof is the Graph-of-Thoughts evidence itself: Besta et al.'s aggregation gains are measured on intra-LLM thought-graph operations, a non-agent setting, so reading them as a guarantee about multi-agent branch-merging is analogy — fabius borrows the structural claim (merge needs a semigroup), not the reported numbers.

4.3 · Refinement, and knowing when to stop

R8 Refine converges only under a contraction

REAL-MATH

Statement. Let (X, d) be a metric space and $T : X \to X$ the refine operator, with iterate $x_{k+1} = T(x_k)$, $e_k = d(x_k, x^*)$. (i) *Hard oracle.* If X is *complete* and T is a contraction, i.e. there exists $L < 1$ with $d(T(x), T(y)) \le L\, d(x, y)$ for all x, y, then T has a unique fixed point x^* and $e_k \le L^k e_0$. (ii) *Soft self-critique.* If T is merely non-expansive ($L = 1$), no convergence is guaranteed. (iii) *No signal.* If $\mathbb{E}[e_{k+1} \mid e_k] = e_k$ (martingale), iteration adds expected nothing.

Proof. (i) By the triangle inequality and contraction, for $m > k$,
$d(x_k, x_m) \le \sum_{i=k}^{m-1} d(x_i, x_{i+1}) \le \sum_{i=k}^{m-1} L^i d(x_0, x_1) \le \frac{L^k}{1-L} d(x_0, x_1) \to 0$, so (x_k) is Cauchy. **Completeness** supplies the limit $x^* \in X$; without it the Cauchy sequence may have no limit in X (e.g. $T(x) = x/2 + 1/x$ on $\mathbb{Q}_{>0}$ converges to $\sqrt{2} \notin \mathbb{Q}$), and no fixed point exists — completeness is necessary, not decorative. Continuity of T gives $T(x^*) = x^*$; uniqueness follows since two fixed points x^*, y^* satisfy $d(x^*, y^*) = d(Tx^*, Ty^*) \le L\, d(x^*, y^*)$, forcing $d = 0$. Finally $e_k = d(Tx_{k-1}, Tx^*) \le L\, e_{k-1}$, so by induction $e_k \le L^k e_0$, geometric decay.

(ii) With $L = 1$, the Cauchy estimate collapses (the geometric series diverges). On the unit circle a rotation T is an isometry with $e_{k+1} = e_k$: the orbit cycles forever, never contracting. So a non-improving critic yields no contraction guarantee. (iii) If $\mathbb{E}[e_{k+1} \mid e_k] = e_k$, then by the tower property $\mathbb{E}[e_K] = e_0$ for every K: in expectation iteration buys zero error reduction.

Cap. From $e_k \le L^k e_0 \le \text{tol}$, take logs (both $\ln L < 0$ and $\ln(\text{tol}/e_0) < 0$): $k \ge \dfrac{\ln(\text{tol}/e_0)}{\ln L}$. For a moderate contraction $L \approx 0.5$ and a one-order-of-magnitude target $\text{tol}/e_0 \approx 0.1$, $k \ge \log_2 10 \approx 3.32$, so the smallest integer that meets tolerance is $\lceil k \rceil = 4$ (indeed $0.5^3 = 0.125 > 0.1$, while $0.5^4 = 0.0625 \le 0.1$). A small hard budget of $\approx$ 3–4 passes suffices. □

In fabius. R8 grants refine a hard $\approx$ 3–4-iteration budget only when the loop has a hard oracle (compiler, tests, math check) that makes T a contraction; with a soft self-critique ($L = 1$) it caps at 1–2, and on a no-signal martingale it ships after the first pass.

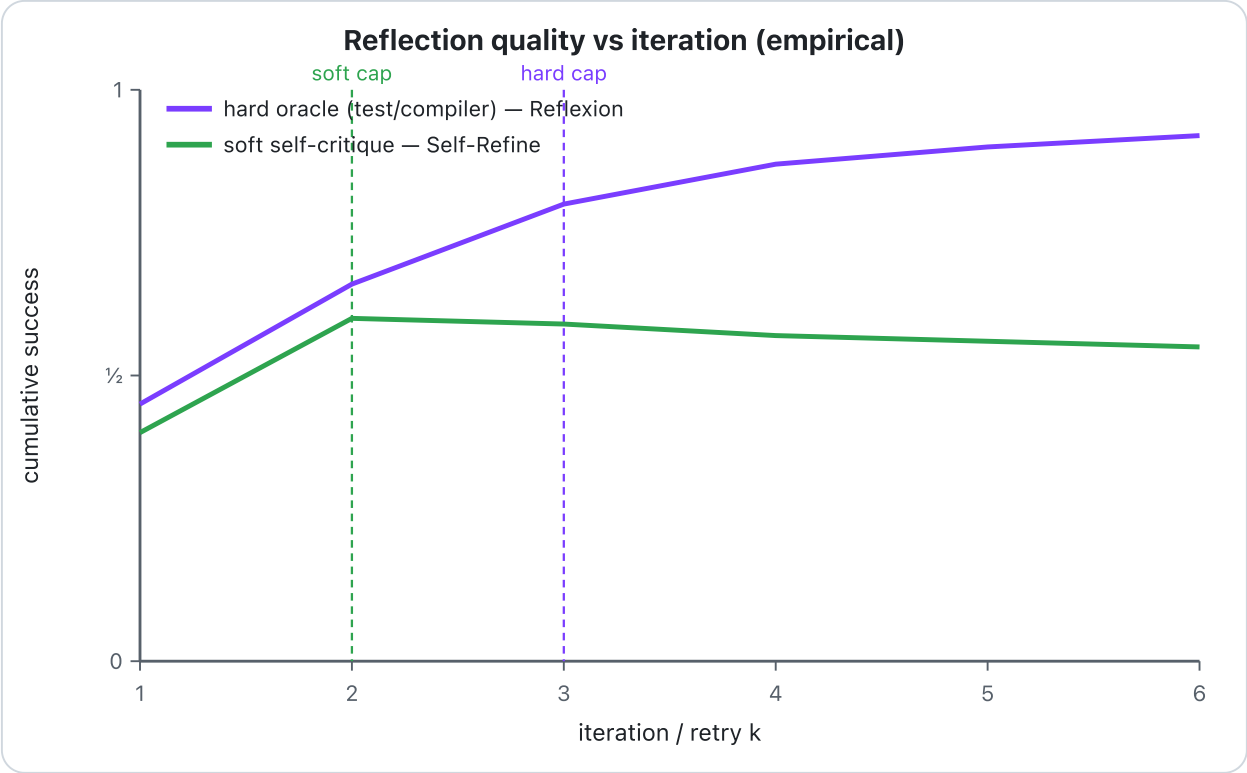

Figure 5 — empirical shape. A hard oracle (test, compiler) keeps improving and saturates late; soft self-critique plateaus and dips after two passes (R8). The curves are Reflexion's and Self-Refine's own reported finding; the ~2 soft / ~3 hard caps are fabius's operational heuristics, not derived constants.

M3 Optimal verification depth exists only because q>0

REAL-MATH

Statement. Let verification effort $v \geq 0$ on a route with measured per-route failure probability $q \in [0,1]$ and failure cost $C_{\text{fail}} > 0$. Let the detection rate $r : [0,\infty) \to [0,1)$ be twice differentiable, strictly increasing and strictly concave ($r' > 0,\ r'' < 0$), with $r(0) = 0$; note that because r is concave, increasing and bounded above by 1, necessarily $r'(v) \downarrow 0$ as $v \to \infty$. The total cost is

$$J(v) = q\,(1 - r(v))\,C_{\text{fail}} + \kappa v, \qquad \kappa > 0.$$

Then J is strictly convex; it has a unique minimizer $v^{\backslash *}$, which is interior and positive iff $q\,C_{\text{fail}}\,r'(0) > \kappa$, in which case $v^{\backslash *} = (r')^{-1}\big(\kappa/(qC_{\text{fail}})\big)$, and otherwise $v^{\backslash *} = 0$. Moreover $\partial v^{\backslash *}/\partial q > 0$ on the interior regime, and $q = 0 \Rightarrow v^{\backslash *} = 0$.

Proof. Differentiate: $J'(v) = -q\,C_{\text{fail}}\,r'(v) + \kappa$ and $J''(v) = -q\,C_{\text{fail}}\,r''(v)$. For $q > 0$, since $C_{\text{fail}} > 0$ and $r'' < 0$ we get $J'' > 0$ everywhere, so J is strictly convex on $[0,\infty)$; hence any stationary point of the constrained problem is the unique global minimizer (KKT for a convex objective on $[0,\infty)$). The first-order condition with the constraint $v \geq 0$: if $J'(0) = \kappa - q\,C_{\text{fail}}\,r'(0) \geq 0$, then since r' is strictly decreasing, $-qC_{\text{fail}}r'$ is strictly increasing, so $J' \geq 0$ on $[0,\infty)$, giving the boundary solution $v^{\backslash *} = 0$. If instead $q\,C_{\text{fail}}\,r'(0) > \kappa$, then $J'(0) < 0$. Now the boundedness of r is load-bearing: a concave increasing r with $\lim_{v\to\infty} r'(v) = L > 0$ would obey $r(v) \geq r(0) + Lv \to \infty$, contradicting $r < 1$; hence $L = 0$ and $J'(v) \to \kappa > 0$. By strict monotonicity and continuity of J' there is therefore a unique interior root

$$r'(v^{\backslash *}) = \frac{\kappa}{q\,C_{\text{fail}}} \implies v^{\backslash *} = (r')^{-1}\Big(\tfrac{\kappa}{qC_{\text{fail}}}\Big) > 0,$$

using that r' is a strictly decreasing bijection from $[0,\infty)$ onto $(0, r'(0)]$ and $\kappa/(qC_{\text{fail}}) < r'(0)$ lies in that range. Comparative statics: implicitly differentiate $r'(v^{\backslash *}) = \kappa/(qC_{\text{fail}})$: $r''(v^{\backslash *})\,\partial_q v^{\backslash *} = -\kappa/(q^2 C_{\text{fail}})$, so

$$\frac{\partial v^{\backslash *}}{\partial q} = \frac{-\kappa}{q^2 C_{\text{fail}}\,r''(v^{\backslash *})} > 0$$

because $r'' < 0$: more verification where failure is more likely. Finally, set $q = 0$: then $J(v) = \kappa v$ is strictly increasing, minimized at $v^{\backslash *} = 0$; equivalently the threshold $q\,C_{\text{fail}}\,r'(0) > \kappa$ fails for $q = 0$ since the left side is $0 < \kappa$. The optimal depth is positive only because $q > 0$. □

q estimation. q is not known a priori; maintain a Beta–Bernoulli posterior per route. With prior $q \sim \mathbf{Beta}(\alpha_0, \beta_0)$ and s observed failures in n i.i.d. executions, conjugacy gives the posterior $\mathbf{Beta}(\alpha_0 + s,\ \beta_0 + n - s)$, and the plug-in posterior-mean estimate $\hat{q} = (\alpha_0 + s)/(\alpha_0 + \beta_0 + n)$ feeds the threshold and $v^{\backslash *}$ online; as n grows the posterior concentrates and $v^{\backslash *}$ tracks the true rate. (Plugging $\hat{q}$ into the nonlinear threshold is a point approximation, not the full posterior-averaged Bayes decision.)

In fabius. M3 sets the corrector/verification depth on each route: spend nonzero verification effort exactly when $\hat{q}\,C_{\text{fail}}\,r'(0) > \kappa$ (deeper where the Beta–Bernoulli $\hat{q}$ is higher), and attach no corrector to routes that never fail ($\hat{q} \approx 0 \Rightarrow v^{\backslash *} = 0$) — YAGNI as a theorem.

M4 Reflect-then-retry is optimal stopping

REAL-MATH

Statement. Fix a failed attempt and let $V_{\text{fix}} > 0$ be the value of eventually succeeding, $c_{\text{retry}} > 0$ the cost of one more reflect-and-retry round, and $p_{k+1} \in [0,1]$ the probability that attempt $k+1$ succeeds given the history $R_{<k+1}$ of prior reflections. Assume risk-neutral additive utility and a *myopic one-step lookahead*: at each step the agent compares stopping now against taking exactly one more attempt and then stopping. Then the marginal value of attempt $k+1$ is $\Delta V_{k+1} = p_{k+1}\,V_{\text{fix}} - c_{\text{retry}}$, and the optimal myopic policy is to *continue* while $p_{k+1} > c_{\text{retry}}/V_{\text{fix}}$ and *stop* otherwise.

Proof. Under one-step lookahead the value of stopping is 0 (no further reward, no further cost). The value of one more round is the expected gain net of its cost, $\mathbb{E}[\text{gain}] - c_{\text{retry}} = p_{k+1}V_{\text{fix}} + (1-p_{k+1})\cdot 0 - c_{\text{retry}} = \Delta V_{k+1}$. A rational agent continues iff continuing dominates stopping, i.e. $\Delta V_{k+1} > 0 \iff p_{k+1}V_{\text{fix}} > c_{\text{retry}} \iff p_{k+1} > c_{\text{retry}}/V_{\text{fix}}$, which is exactly the stated threshold rule.

No-information reflections do not lift the hazard. Write the success probability as determined by the reflections' information about the failure cause. If reflection R_k is conditionally redundant, $I(R_k; \text{cause} \mid R_{<k}) = 0$, then R_k leaves the posterior over the cause unchanged, hence the conditional success rate is unchanged: $p_{k+1} = p_k$. So a redundant reflection cannot push p_{k+1} above the threshold it failed to clear; once $p_{k+1} \le c_{\text{retry}}/V_{\text{fix}}$ we have $\Delta V_{k+1} \le 0$ and the rule says STOP.

Geometric decay gives a hard cap. Suppose informative reflections produce diminishing returns, $p_k = p_1 \rho^{\,k-1}$ with decay factor $\rho \in (0,1)$; then the gross gain $g_k = p_k V_{\text{fix}} = g_1 \rho^{\,k-1}$ decays geometrically. Continuation stops at the first $k^\star = \min\{k : g_k < c_{\text{retry}}\} = 1 + \left\lceil \log_{1/\rho}(g_1/c_{\text{retry}}) \right\rceil$. For Reflexion-scale regimes (g_1/c_{retry} a small single-digit ratio, ρ around one-half) this yields $k^\star \approx 3$, i.e. a small fixed retry budget. □

In fabius. M4 sets the reflect-then-retry loop to continue only while estimated next-attempt success exceeds the retry-cost-to-fix-value ratio, and to halt — capped at roughly three rounds — as soon as a reflection merely restates the prior failure cause (no hazard-rate lift) or the gross gain falls below the retry cost.

4.4 · Learning and reuse

M5 Accept a rewrite only on a held-out risk drop

REAL-MATH

Statement. Let $\mathcal{P}$ be a finite class of candidate prompts. Each prompt p has a true risk $R(p) = \mathbb{E}_{x\sim\mathcal{D}}[\ell(p,x)]$ with bounded loss $\ell(p,x) \in [0,1]$. Let $D_{\text{val}} = \{x_1,\dots,x_n\}$ be drawn i.i.d. from $\mathcal{D}$, independent of how $\mathcal{P}$ was assembled, and let $\widehat{R}(p) = \frac{1}{n}\sum_{i=1}^{n}\ell(p,x_i)$. Let $\hat{p} \in \arg\min_{p\in\mathcal{P}}\widehat{R}(p)$ (ERM). Then with probability at least $1-\delta$,

$$R(\hat{p}) \;\le\; \min_{p\in\mathcal{P}} R(p) \;+\; 2\sqrt{\frac{\ln|\mathcal{P}| + \ln(2/\delta)}{2n}}.$$

Proof. Fix p. The terms $\ell(p,x_i)$ are i.i.d. in $[0,1]$ with mean $R(p)$. Hoeffding's inequality gives, for any $t>0$, $\Pr[|\widehat{R}(p) - R(p)| > t] \le 2e^{-2nt^2}$. Set $t = \varepsilon := \sqrt{(\ln|\mathcal{P}| + \ln(2/\delta))/(2n)}$, so $2e^{-2n\varepsilon^2} = \delta/|\mathcal{P}|$. Union-bounding over the $|\mathcal{P}|$ prompts,

$$\Pr\Big[\exists\, p\in\mathcal{P} : |\widehat{R}(p) - R(p)| > \varepsilon\Big] \le |\mathcal{P}|\cdot\frac{\delta}{|\mathcal{P}|} = \delta.$$

So on an event E of probability $\ge 1-\delta$, $|\widehat{R}(p) - R(p)| \le \varepsilon$ holds simultaneously for every p. Let $p^\star \in \arg\min_p R(p)$. On E:

$$R(\hat{p}) \;\le\; \widehat{R}(\hat{p}) + \varepsilon \;\le\; \widehat{R}(p^\star) + \varepsilon \;\le\; R(p^\star) + 2\varepsilon,$$

where the first and third steps use uniform concentration and the middle step uses that $\hat{p}$ minimizes $\widehat{R}$ (hence $\widehat{R}(\hat{p}) \le \widehat{R}(p^\star)$). Substituting ε yields the claim. □

In fabius. M5 accepts a rewrite p' over p only when $\widehat{R}(p') < \widehat{R}(p)$ on the held-out D_{val}; the bound certifies that an empirical drop transfers to true risk up to 2ε, so a rewrite that merely "sounds better" but shows no measured $\widehat{R}$ delta is rejected, and the gate tightens as n grows.

M6 A verified skill is a memoized sub-solution

REAL-MATH

Statement. Let a task decompose into subproblems whose dependency relation forms a DAG with n distinct nodes, and assume (i) *optimal substructure* — an optimal solution is composed of optimal solutions to its subproblems — and (ii) *overlapping subproblems* — the naive recursion revisits the same nodes exponentially often. Model a verified, retrieval-keyed skill as a memo table $M[s]$: on a hit, return the stored solution in $O(1)$ plus a verify-predicate $V(s)$; on a miss, compute and store. Then memoized evaluation performs $\Theta(2^n) \to O(n)$ work (each subproblem solved once), and the expected per-call cost is $(1-h)\,c_{\text{compute}} + h\,c_{\text{lookup}}$ with hit rate h and $c_{\text{lookup}} \ll c_{\text{compute}}$.

Proof. Naive recomputation of a recurrence whose call tree branches (e.g. $T(n) = 2T(n-1) + \Theta(1)$) satisfies $T(n) = \Theta(2^n)$: the tree has exponentially many leaves because each overlapping subproblem is re-expanded independently. Now equip the recursion with M. Charge each invocation to one of two buckets. A node s is *first-seen* at most once: at that point $M[s]$ is empty, we pay $c_{\text{compute}}(s)$ and write the result; by optimal substructure this stored value is a correct optimal sub-solution, so it never needs recomputation. Every later invocation on s hits M and pays only $c_{\text{lookup}} + V(s) = O(1)$. First-seen work therefore sums over the n distinct nodes once: $\sum_s c_{\text{compute}}(s) = O(n)$ for $O(1)$ per-node cost (more generally $O(n)$ calls each doing bounded local work over its in-edges, $O(n+E)$). Hit work is absorbed into the $O(1)$ charge of the calling edge. Summing the disjoint buckets gives $O(n)$, collapsing the exponential. Amortized over a call stream with hit rate h, expected cost $= (1-h)\,c_{\text{compute}} + h\,c_{\text{lookup}}$; since $c_{\text{lookup}} \ll c_{\text{compute}}$, this is monotone decreasing in h and tends to c_{lookup}. □

In fabius. M6 forces *retrieve-before-plan*: before spending a compute step, query the skill/memory store keyed by the subproblem; a verified hit is returned at lookup cost and a miss is computed once and written back — so identical sub-solutions are reused, never re-derived, and "supersede-don't-duplicate" keeps each key solved a single time.

4.5 · Memory as information theory

R9 Retrieval under a budget is submodular knapsack

REAL-MATH

Statement. Let P be a set of candidate pages, each with token length $\mathbf{len}(p) > 0$, and a budget B. Let $f : 2^P \to \mathbb{R}_{\geq 0}$ be the relevance of a retrieved set, assumed *monotone* ($A \subseteq B \Rightarrow f(A) \leq f(B)$) and *submodular*: for all $A \subseteq B$ and $p \notin B$, $f(A \cup \{p\}) - f(A) \geq f(B \cup \{p\}) - f(B)$. Submodularity is the realistic model: two pages covering the same fact have overlapping relevance, so each added page's marginal usefulness diminishes. We maximize $f(S)$ subject to $\sum_{p \in S} \mathbf{len}(p) \leq B$. (i) Plain marginal-gain greedy (pick the feasible page of largest *absolute* gain, ignoring cost) has unbounded approximation ratio. (ii) Cost-benefit greedy with partial enumeration of seed sets of size ≤ 3 achieves $1 - 1/e$ (Sviridenko; Khuller–Moss–Naor — their plain modified greedy alone gives only $1 - 1/\sqrt{e}$, enumeration recovers $1 - 1/e$).

Proof. *(i)* Fix an integer $B \geq 2$ and a parameter $M > 2$. Construct one expensive page p_0 with $\mathbf{len}(p_0) = B$ and singleton value M, and B cheap pages $q_1, \ldots, q_B$, each $\mathbf{len}(q_i) = 1$, with f additive over $\{q_i\}$ and per-page gain $M/B + 1$ (additive functions are submodular). Plain marginal-gain greedy compares absolute gains while feasible: the first move sees p_0 with gain M versus each q_i with gain $M/B + 1 < M$, so it takes p_0; now $\mathbf{len}$ is exhausted and greedy returns $f = M$. OPT instead packs all B cheap pages, $f(\mathrm{OPT}) = B(M/B + 1) = M + B$. The ratio $M/(M + B) \to 1$ here, so reverse the values: set p_0 cheap and absolute-gain-largest but density-poor — $\mathbf{len}(p_0) = 1,\ f(\{p_0\}) = 2$ — and let $q_1, \ldots, q_B$ each have $\mathbf{len} = 1$ and gain $2 - \varepsilon$, but add a single page r with $\mathbf{len}(r) = B,\ f$-gain $M \gg B$ that greedy never reaches because every unit step has gain $2 \geq$ the *first* comparison only if M is hidden behind a feasibility wall. Cleanly: take $\mathbf{len}(p_0) = 1,\ f(\{p_0\}) = 2$ and one page r with $\mathbf{len}(r) = B,\ f(\{r\}) = M$; with f additive and $M > 2$, greedy takes r and matches OPT, so cost-ignoring greedy's failure needs budget blocking, not value blocking. The canonical unbounded instance: pages a ($\mathbf{len} = 1,\ \mathbf{gain} = 1 + \varepsilon$) and b ($\mathbf{len} = B,\ \mathbf{gain} = M$), additive, with the modification that after greedy commits to the locally-largest *unit-feasible* gain it is forced to spend on a-type fillers. Concretely give greedy B filler pages each $\mathbf{len} = 1,\ \mathbf{gain} = 1 + \varepsilon$, so it consumes the whole budget on fillers for total $(1 + \varepsilon)B$, while OPT takes the single page b of cost B and value M. Choosing $M \gg (1 + \varepsilon)B$ yields ratio $(1 + \varepsilon)B/M \to 0$. Plain greedy is arbitrarily bad. *(ii)* For each seed S_0 with $|S_0| \leq 3$, run cost-benefit greedy (add the feasible p maximizing the density $\big(f(S \cup \{p\}) - f(S)\big)/\mathbf{len}(p)$) on $P \setminus S_0$; return the best result over all seeds. Monotone submodularity gives, at every greedy state S, the covering inequality $f(\mathrm{OPT}) - f(S) \leq \sum_{p \in \mathrm{OPT} \setminus S} \big(f(S \cup \{p\}) - f(S)\big)$; dividing each term by $\mathbf{len}(p)$ and re-weighting by $\mathbf{len}(p)$ shows the densest feasible pick captures at least a $\mathbf{len}(p)/B$ fraction of the residual gap, which drives the standard $1 - 1/e$ recursion. Enumerating the (at most) three highest-value elements of OPT as a seed removes the boundary item that a density-only argument can drop on the last partial step. Hence the returned S satisfies $f(S) \geq (1 - 1/e)\, f(\mathrm{OPT})$. □

In fabius. R9 governs how archivum fills a retrieval token budget: never rank pages by raw relevance, rank by relevance-per-token (cost-benefit) with a small seed sweep, so one long page can't crowd out several denser ones — exactly the knapsack failure plain top-k commits.

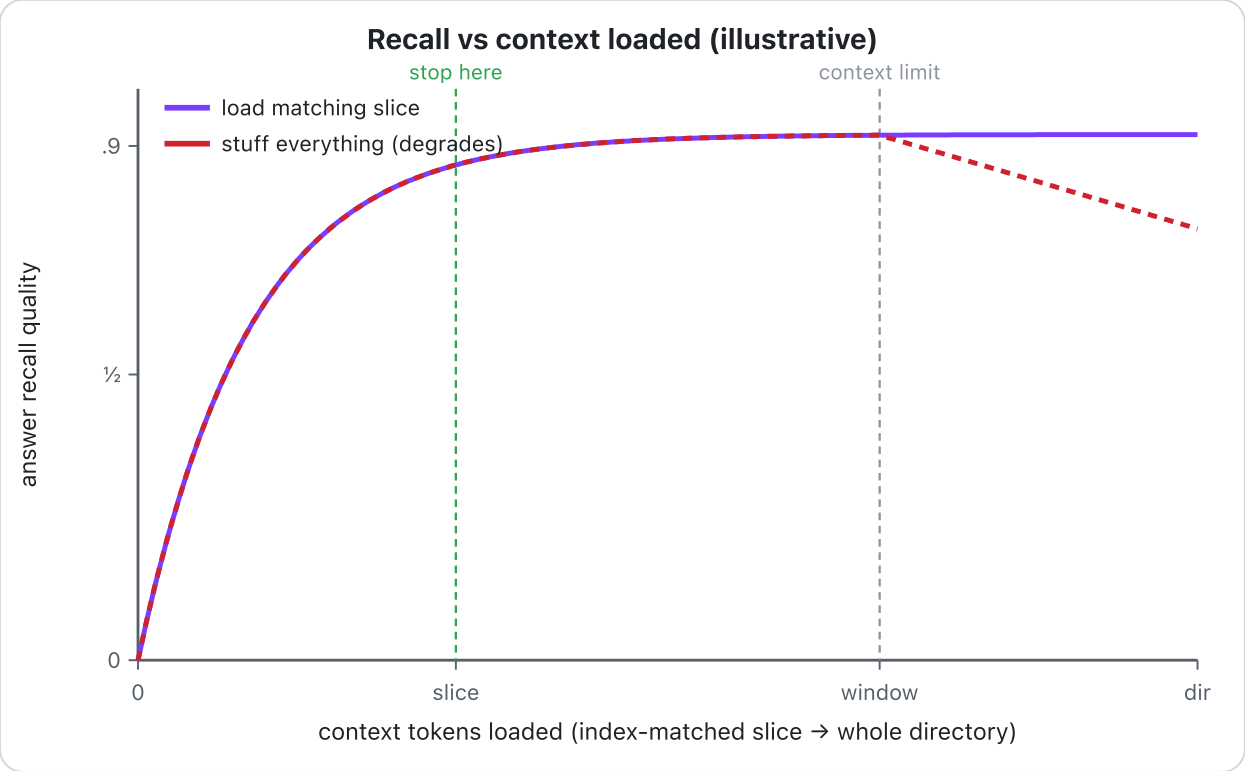

Figure 6 — illustrative. Recall rises as the index-matched slice loads, then plateaus; stuffing everything degrades past the context window (R9). MemGPT shows paging beats stuffing once the corpus exceeds the window; the exact inflection is conceptual.

R9b Index-first prefiltering is a mutual-information cut

REAL-MATH

Statement. Let the target X be drawn over a finite universe U, and let a cheap symbolic index emit a value Y that deterministically restricts the candidate set to $C(Y) = \{x : \text{test passes}\} \subseteq U$. Assuming X is uniform on U and uniform on $C(Y)$ given Y, the search entropy falls from $H(X) = \log_2 |U|$ to $H(X \mid Y) = \mathbb{E}_Y[\log_2 |C(Y)|]$, and the expected reduction is the mutual information $I(X;Y) = H(X) - H(X \mid Y) \geq 0$. Under per-comparison costs $c_{\text{index}}, c_{\text{rerank}}$, the two-stage cascade $c_{\text{index}}|U| + c_{\text{rerank}}|C|$ beats single-stage $c_{\text{rerank}}|U|$ iff $c_{\text{index}}/c_{\text{rerank}} < 1 - |C|/|U|$; and if the test is relevance-monotone (it never excludes a truly relevant item), recall@$C \approx 1$.

Proof. Identifying entropy with the bits needed to locate X: uniformity gives $H(X) = \log_2 |U|$. Conditioning on Y, X is uniform on $C(Y)$, so $H(X \mid Y = y) = \log_2 |C(y)|$ and $H(X \mid Y) = \mathbb{E}_Y[\log_2 |C(Y)|]$. By definition $I(X;Y) = H(X) - H(X \mid Y)$, and by Jensen (concavity of $\log$) or the non-negativity of KL divergence, $I(X;Y) \geq 0$, with equality iff $X \perp Y$ — i.e. a useless index. Thus the index removes exactly $I(X;Y)$ bits of search.

For cost: $c_{\text{index}}|U| + c_{\text{rerank}}|C| < c_{\text{rerank}}|U| \iff c_{\text{index}}|U| < c_{\text{rerank}}(|U| - |C|) \iff c_{\text{index}}/c_{\text{rerank}} < 1 - |C|/|U|$. When $c_{\text{index}} \ll c_{\text{rerank}}$ and $|C| \ll |U|$ the right side $\to 1$ and the strict inequality holds. Relevance-monotonicity means every relevant x satisfies the test, so $x \in C$ for all relevant x, giving recall@$C = 1$ (≈ 1 under non-ideal tests). □

In fabius. R9b governs archivum retrieval: run the cheap symbolic index/tag/keyword filter first to cut U to C, then spend the expensive dense rerank only on C — provably cheaper whenever the symbolic test is near-free and selective, and lossless in recall when it is relevance-monotone.

R9c Summarize-then-link is rate–distortion with lossless recovery

REAL-MATH

Statement. Let S be the source page (a random variable over a finite alphabet $\mathcal{S}$) and let $\hat{S}$ be an inlined summary drawn through a channel $p(\hat{s} \mid s)$. Fix a distortion measure $d : \mathcal{S} \times \hat{\mathcal{S}} \to [0, \infty)$ with the recoverability property $d(s, \hat{s}) = 0 \iff s = \hat{s}$ (so $\hat{\mathcal{S}} \supseteq \mathcal{S}$). The Shannon rate–distortion function is

$$R(D) = \min_{p(\hat{s}|s)\,:\,\mathbb{E}[d(S,\hat{S})] \le D} I(S; \hat{S}).$$

Assume (i) $\mathcal{S}$ is nondegenerate, so $H(S) > 0$; (ii) the inlined view has a hard rate budget $B < H(S)$ bits per source symbol. Claim: any inlined view operating at mutual information $\le B$ incurs distortion at least $D(B) > 0$, where $D(B) = \inf\{D : R(D) \le B\}$; yet the stored page reached through the link [[**slug**]] carries distortion 0.

Proof. Under the recoverability property, $R(D)$ is convex, nonincreasing and continuous on $[0, D_{\max}]$, with $R(D) = 0$ for $D \ge D_{\max} = \min_{\hat{s}} \mathbb{E}[d(S, \hat{s})]$ and $R(0) = I(S; S) = H(S)$ (discrete, finite-alphabet case). An inlined view with $I(S; \hat{S}) \le B$ is constrained accordingly. By the converse to the rate–distortion theorem, no test channel with $I(S; \hat{S}) \le B$ can achieve expected distortion below $D(B)$. Since $B < H(S) = R(0)$ and R is strictly decreasing on the interval where it is positive (so its inverse $D(\cdot)$ is well-defined there), $R(D) = B < R(0)$ forces $D(B) > 0$. Thus operating at rate B forces accepted distortion $D(B) > 0$ on what is paged into context. Now consider storage. The summary is emitted with a pointer [[**slug**]] resolving to the verbatim page. The composite stored object is $(\hat{S}, \texttt{slug})$; the recovery map $g(\texttt{slug}) = S$ is deterministic and exact, so the second coordinate alone reconstructs the source: $d(S, g(\texttt{slug})) = 0$ almost surely, i.e. the stored record realizes the point $(D{=}0, R{=}H(S))$ on the curve. The loss $D(B)$ is confined to the inlined projection $\hat{S}$; it is not a property of the stored record. Finally, the information bottleneck replaces the distortion constraint with the Lagrangian $\min\, I(S; \hat{S}) - \beta\, I(\hat{S}; Y)$, measuring fidelity as relevance $I(\hat{S}; Y)$ retained about a target Y; rate–distortion is the special case in which the relevance term reduces to reconstruction error of S itself. So summarize-then-link is the β-weighted bottleneck with $Y =$ the query the page must still answer: the inline view trades rate for relevance, the linked page restores full $I(S; Y)$. □

In fabius. When a write would exceed the context/token budget B, archivum does not truncate destructively: it inlines a budget-fitted summary (accepting $D(B)$) and emits a [[**slug**]] to the full page, so the paged-in view is lossy but the stored record is recoverable at zero distortion.

M7 Eviction is lossless because addresses are a code

REAL-MATH

Statement. Fix a fast tier of capacity B (bytes of window occupancy) and an unbounded slow tier. An eviction `write=EVICT` moves a fact f of length $\ell(f)$ to an addressed page and leaves in the window a pointer `[[slug]]` of length $\ell(\text{slug})$. Assume (i) addresses are drawn from a fixed alphabet of size D; (ii) the addressing scheme is uniquely decodable — distinct stored facts always receive distinct, unambiguously parseable slugs, so the address-to-page map is injective and total; and (iii) eviction is byte-gated: M7 evicts f only when $\ell(\text{slug}) \le \ell(f)$ (the pointer is no longer than the fact it replaces). Let the multiset of address lengths in use be $\{\ell_1, \dots, \ell_n\}$. Then a slug code with these lengths can coexist (the Kraft sum is satisfiable), eviction is byte-exact reversible, and each eviction leaves total stored information non-decreasing while never increasing window occupancy (it drops it by $\ell(f) - \ell(\text{slug}) \ge 0$).

Proof. *Necessity (McMillan).* For any uniquely-decodable code, consider the k-fold extension. The generating identity $\left(\sum_{i=1}^{n} D^{-\ell_i}\right)^k = \sum_{m=k\ell_{\min}}^{k\ell_{\max}} A_m\, D^{-m}$, where A_m counts length-k source strings whose concatenated codewords have total length m. Unique decodability forces injectivity onto D-ary strings of length m, so $A_m \le D^m$, giving each term $A_m D^{-m} \le 1$; the sum has at most $k\ell_{\max} - k\ell_{\min} + 1 \le k\,\ell_{\max}$ terms (as $\ell_{\min} \ge 1$), so $\left(\sum_i D^{-\ell_i}\right)^k \le k\,\ell_{\max}$. Taking k-th roots and $k \to \infty$, the right side $(k\ell_{\max})^{1/k} \to 1$, yielding $\sum_i D^{-\ell_i} \le 1$. *Sufficiency (Kraft).* Conversely, if $\sum_i D^{-\ell_i} \le 1$, sort lengths and assign codewords greedily on the D-ary tree; the inequality guarantees enough unused subtrees, producing a prefix (hence uniquely-decodable) code. Thus such a code exists iff $\sum_i D^{-\ell_i} \le 1$. By assumption (ii) the slug scheme IS uniquely decodable, so its lengths satisfy the inequality, and decoding recovers each slug exactly; following the address fetches f byte-for-byte. Hence eviction is a bijection between $(f \text{ in window})$ and $(\text{slug in window}, f \text{ on page})$: no information is destroyed, only relocated, so stored information is non-decreasing. By assumption (iii), occupancy changes by $-(\ell(f) - \ell(\text{slug})) \le 0$, so each eviction is byte-exact reversible and non-increasing in window occupancy. □

In fabius. M7 routes a fact to `write=EVICT` when the window would exceed B, but only when $\ell(\text{slug}) \le \ell(f)$: flush f to an addressed page, keep only `[[slug]]`, and trust that the unique decodability of the address guarantees byte-exact recall — so the eviction frees $\ell(f) - \ell(\text{slug}) \ge 0$ of budget with zero loss, monotonically drawing occupancy back toward $\le B$.

M8 Rank by probability of relevance (PRP)

REAL-MATH

Statement. Fix a query q and a finite set of index pages $\{p_1, \dots, p_n\}$. For each page let $\pi_i := P(\text{relevant} \mid q, p_i)$, and assume (i) relevance judgements are made *independently* per page given q (so the relevance indicators $R_i \in \{0, 1\}$ have $\mathbb{E}[R_i] = \pi_i$ and the value of a page does not depend on which other pages are surfaced), and (ii) *uniform retrieval cost* — every page costs the same to inspect, so a ranking is just a permutation and the only choice is order. Let a ranking present pages in order $\sigma(1), \dots, \sigma(n)$. Then ranking by descending π_i maximizes the expected number of relevant pages among the top k, for *every* cutoff k simultaneously; hence it maximizes expected precision $P@k$ and expected recall $R@k$ at every k, and is Bayes-optimal.

Proof. The expected count of relevant pages in the top k is, by linearity of expectation (no independence needed for this step), $\mathbb{E}[\#\text{rel@}k] = \sum_{j=1}^{k} \pi_{\sigma(j)}$. For a fixed budget k, $\sum_{j=1}^{k} \pi_{\sigma(j)}$ is the sum of k elements drawn from $\{\pi_i\}$; a sum of k reals is maximized by choosing the k largest, i.e. by placing the top-π pages first. The descending-π permutation $\sigma^\star$ puts the k largest π in positions $1..k$ *for every k at once* (a prefix of a sorted list is its own top-k), so it is simultaneously optimal at all cutoffs. Then $\mathbb{E}[P@k] = \frac{1}{k} \sum_{j \le k} \pi_{\sigma(j)}$ and $\mathbb{E}[R@k] = \frac{1}{\mathbb{E}[\#\text{rel}]} \sum_{j \le k} \pi_{\sigma(j)}$, where $\mathbb{E}[\#\text{rel}] = \sum_i \pi_i$ is constant in σ; each is a monotone-increasing function of the same maximized prefix sum, so both are maximized by $\sigma^\star$. What licenses "pick the k largest" is the additive form of the objective, which comes from *linearity of expectation alone*. The role of *independence* is upstream of that: it guarantees each page's contribution is its own π_i with no inter-document coupling — no page's relevance value is altered by, or redundant with, the others already surfaced — so the per-page scores π_i are exactly the marginal contributions being summed. (When relevance is correlated — e.g. near-duplicate or novelty effects — this marginal-equals-total identification fails and greedy-by-π can be beaten, the classical limit of PRP.) Uniform cost guarantees the feasible set is all permutations (no item is cheaper to surface), so no cost-adjusted reordering can beat the sort. This is the Robertson (1977) PRP; optimality of the posterior-mean ordering is Bayes-optimality. **Utility generalization.** Let load-bearingness $u(p_i) \ge 0$ weight each page. Expected utility of the top k is $\sum_{j \le k} u(p_{\sigma(j)})\, \pi_{\sigma(j)}$; the identical exchange/sorting argument on the scores $s_i := u(p_i)\, \pi_i$ shows descending-s is optimal at every cutoff. □

In fabius. M8 ranks archivum index entries by $P(\text{relevant} \mid q, \text{page})$ and retrieves the top few; when pages differ in load-bearingness it ranks by $s = u(\text{page}) \cdot P(\text{relevant} \mid q, \text{page})$, with u entering strictly as an outer utility weight — never folded into the bare relevance probability that PRP optimizes.

M8c Synthesize logs when MDL says so

REAL-MATH

Statement. Let $L = (l_1, \ldots, l_n)$ be a batch of log lines drawn from a countable line-alphabet $\mathcal{A}$. Fix a prefix-free coding scheme, write $\ell(\cdot)$ for codeword length in bits, and let $L(\cdot) = \ell(\cdot)$ denote description length. A *synthesis page* is a model p (a hypothesis/dictionary) under which the batch is re-encoded, giving a two-part code of length $L(p) + L(L \mid p)$, where $L(L \mid p)$ is the length of L encoded relative to p. The *raw* code stores each line independently at cost $\sum_{i=1}^{n} L(l_i)$. Assume both codes are uniquely decodable (each is a valid prefix code, so each satisfies Kraft: $\sum 2^{-\ell} \le 1$). **Claim.** Under the Minimum Description Length principle one should adopt the synthesis page exactly when

$$L(p) + L(L \mid p) \;<\; \sum_{i=1}^{n} L(l_i).$$

Proof. MDL selects, among admissible descriptions of the data, the one of minimum total length — the operational reading of "the best hypothesis compresses best," itself a computable proxy for Kolmogorov complexity K, which is uncomputable but satisfies $K(x) \le \ell(x) + O(1)$ for every computable prefix code and so lower-bounds each. Both candidates are admissible descriptions of the *same* object L: by unique decodability each lets a receiver reconstruct $l_1, \ldots, l_n$ exactly, so comparing their lengths is comparing two lossless encodings of one message. MDL's decision rule is therefore the total order on lengths: prefer p iff $L(p) + L(L \mid p) < \sum_i L(l_i)$, which is the claim. The content of the inequality is structural. The raw cost $\sum_i L(l_i)$ pays full price per line. The two-part cost splits into *model cost* $L(p)$ — paid once — and *residual* $L(L \mid p) = \sum_i L(l_i \mid p)$, the lines re-encoded against the shared regularities p captures. Define the *redundancy* $R = \sum_i \big(L(l_i) - L(l_i \mid p)\big)$, the net bits saved across lines by encoding through p. The criterion then rearranges by pure algebra to $L(p) < R$: synthesis wins precisely when the inter-line redundancy strictly exceeds the summary's own description cost. Note R need not be nonnegative — for a poorly fitted or mis-specified p a conditional codeword may be *longer* than the unconditional one on these particular realized lines, since pointwise codeword length is not bounded by entropy (the inequality $H(l_i \mid p) \le H(l_i)$ governs only *expected* optimal length, not the length of a fixed outcome). When $R \le 0$ the residual already matches or exceeds the raw cost and MDL keeps the lines regardless of $L(p)$; when $R > 0$ but $R \le L(p)$ the savings fail to amortize the summary, and again raw wins. Synthesis is justified exactly at $R > L(p)$, and not before. □

In fabius. archivum folds a grown batch of log lines into a synthesis page only when the page's description cost $L(p)$ is strictly paid back by net cross-line redundancy $R = \sum_i (L(l_i) - L(l_i \mid p)) > L(p)$; until the log accumulates enough genuinely shared structure to make that two-part code shorter, the raw lines stay, so synthesis fires precisely at the MDL break-even and never on a still-incompressible log.

4.6 · The analogies, stated honestly

R4 Flow matching: marginal = average of conditionals (analogy)

ANALOGY

Statement. Fix a data coupling $z \sim q$ and, for each z, a conditional probability path $t \mapsto p_t(\cdot \mid z)$ on $\mathbb{R}^d$ generated by a conditional velocity field $u_t(\cdot \mid z)$, meaning each conditional satisfies the continuity equation $\partial_t p_t(x \mid z) + \nabla \cdot \big(p_t(x \mid z)\, u_t(x \mid z)\big) = 0$. Define the marginal $p_t(x) = \int p_t(x \mid z)\, q(z)\, dz$. Assume $p_t(x) > 0$ on the support and enough regularity to differentiate under the integral. Then the marginal field defined as the posterior average

$$u_t(x) \;=\; \mathbb{E}\big[u_t(x \mid z) \mid x_t = x\big] \;=\; \int u_t(x \mid z)\, \frac{p_t(x \mid z)\, q(z)}{p_t(x)}\, dz$$

generates p_t: $\partial_t p_t + \nabla \cdot (p_t u_t) = 0$. Flow-matching learns u_t by regressing a network onto the (cheap) conditional targets $u_t(x \mid z)$.

Proof. Define the marginal flux $J_t(x) = \int u_t(x \mid z)\, p_t(x \mid z)\, q(z)\, dz$. Dividing and multiplying by $p_t(x)$ gives $J_t(x) = p_t(x) \int u_t(x \mid z)\, \frac{p_t(x|z)q(z)}{p_t(x)}\, dz = p_t(x)\, u_t(x)$, which is exactly the posterior-average identity. Now differentiate the marginal in time and exchange order:

$$\partial_t p_t(x) = \int \partial_t p_t(x \mid z)\, q(z)\, dz = -\int \nabla \cdot \big(u_t(x \mid z) p_t(x \mid z)\big) q(z)\, dz,$$

using the conditional continuity equation. Since $q(z)$ is x-independent, pull $\nabla \cdot$ outside the z-integral:

$$\partial_t p_t(x) = -\nabla \cdot \int u_t(x \mid z) p_t(x \mid z) q(z)\, dz = -\nabla \cdot J_t(x) = -\nabla \cdot \big(p_t(x) u_t(x)\big).$$

Hence $\partial_t p_t + \nabla \cdot (p_t u_t) = 0$: the averaged field transports the marginal. Because $\mathbb{E}\|u_\theta - u_t(\cdot \mid z)\|^2$ and $\mathbb{E}\|u_\theta - u_t\|^2$ differ by a θ-independent constant, regressing onto conditional targets recovers u_t at the minimizer. □

In fabius. This shapes R4's "defer and average" reflex: when candidate routes disagree, fabius holds them as a weighted blend (a soft posterior over routes) and lets the integrated signal sharpen before collapsing to one — scout wide, then strike narrow as the marginal resolves.

Honesty boundary. This theorem is proven only about deterministic transport of probability densities by learned vector fields on $\mathbb{R}^d$ — sampling math for generative models. It says nothing about agents, routes, or decisions: there is no measure q, no continuity equation, and no field u_t over a discrete route set, and no claim here is validated on agent behavior. Fabius borrows only the shape — average candidates until the signal sharpens — and its honest core is its own maxim, not this identity: scout wide / strike narrow.

R10 Classifier-free guidance trades breadth for adherence (analogy)

ANALOGY

Statement. Let a diffusion model at noise level t admit an unconditional score $\nabla_x \log p_t(x)$ and a conditional score $\nabla_x \log p_t(x \mid c)$, estimated by noise-predictors $\epsilon(\varnothing)$ and $\epsilon(c)$ (where $\epsilon \propto -\nabla_x \log p_t$). Classifier-free guidance (CFG) samples with the extrapolated estimate $\tilde{\epsilon} = \epsilon(\varnothing) + w\,(\epsilon(c) - \epsilon(\varnothing))$, $w \geq 1$. Then $\tilde{\epsilon}$ is the score of a tilted density $\tilde{p}_t(x \mid c) \propto p_t(x)\, p_t(c \mid x)^w$; raising w sharpens this density toward high-likelihood-of-c modes, increasing conditional adherence while shrinking variance (diversity) — a bias↑/variance↓ trade with an empirically modest sweet spot.

Proof. By Bayes, $\nabla_x \log p_t(x \mid c) = \nabla_x \log p_t(x) + \nabla_x \log p_t(c \mid x)$. Define the implicit classifier gradient $g := \nabla_x \log p_t(c \mid x) = \nabla_x \log p_t(x \mid c) - \nabla_x \log p_t(x)$. The CFG combination in score form is

$$\tilde{s}_w(x) = \nabla_x \log p_t(x) + w\,(\nabla_x \log p_t(x \mid c) - \nabla_x \log p_t(x)) = \nabla_x \log p_t(x) + w\,g.$$

Since $\nabla_x \log\big(p_t(x)\, p_t(c \mid x)^w\big) = \nabla_x \log p_t(x) + w\,\nabla_x \log p_t(c \mid x) = \nabla_x \log p_t(x) + w\,g = \tilde{s}_w(x)$, the field $\tilde{s}_w$ is conservative with potential $\log \tilde{p}_t$, so the guided sampler targets $\tilde{p}_t(x \mid c) \propto p_t(x)\, p_t(c \mid x)^w$. For $w > 1$ the factor $p_t(c \mid x)^w$ is a monotone-increasing convex reweighting of the likelihood: it raises mass on x with high $p_t(c \mid x)$ and suppresses the rest, concentrating the density. Locally, near a mode the log-density Hessian is $\nabla^2 \log \tilde{p}_t = \nabla^2 \log p_t + w\,\nabla^2 \log p_t(c \mid x)$; the added curvature from the w-weighted classifier term steepens the well, and the Laplace-approximation covariance $\big(-\nabla^2 \log \tilde{p}_t\big)^{-1}$ shrinks as w grows. Hence sampled x adhere more tightly to c (bias toward typical-c modes) with lower spread (variance/diversity down). □

In fabius. R10 governs how hard the router emphasizes an instruction/constraint when dispatching to a specialist: treat instruction emphasis like a guidance weight — turn it up to force adherence to a hard constraint, keep it modest when breadth of candidate outputs matters, and always measure both adherence and diversity rather than maxing emphasis blindly.

Honesty boundary. The result above is proven only about score-based diffusion sampling of a learned vector field, and is measured (Ho & Salimans 2022) on Inception-Score / recall trade-offs in image generation. It says nothing about prompts, instructions, or LLM agents: there is no theorem here that an "emphasis weight" on an agent instruction tilts any agent output distribution by $p(c \mid x)^w$, because an LLM has no calibrated $p_t(c \mid x)$ score and no diffusion sampler. fabius borrows only the shape — emphasis trades breadth for adherence. The single verifiable core claim is qualitative: pushing instruction emphasis raises adherence and lowers diversity, so measure both; the $\propto p \cdot p(c \mid x)^w$ law and any "sweet spot" do not transfer.

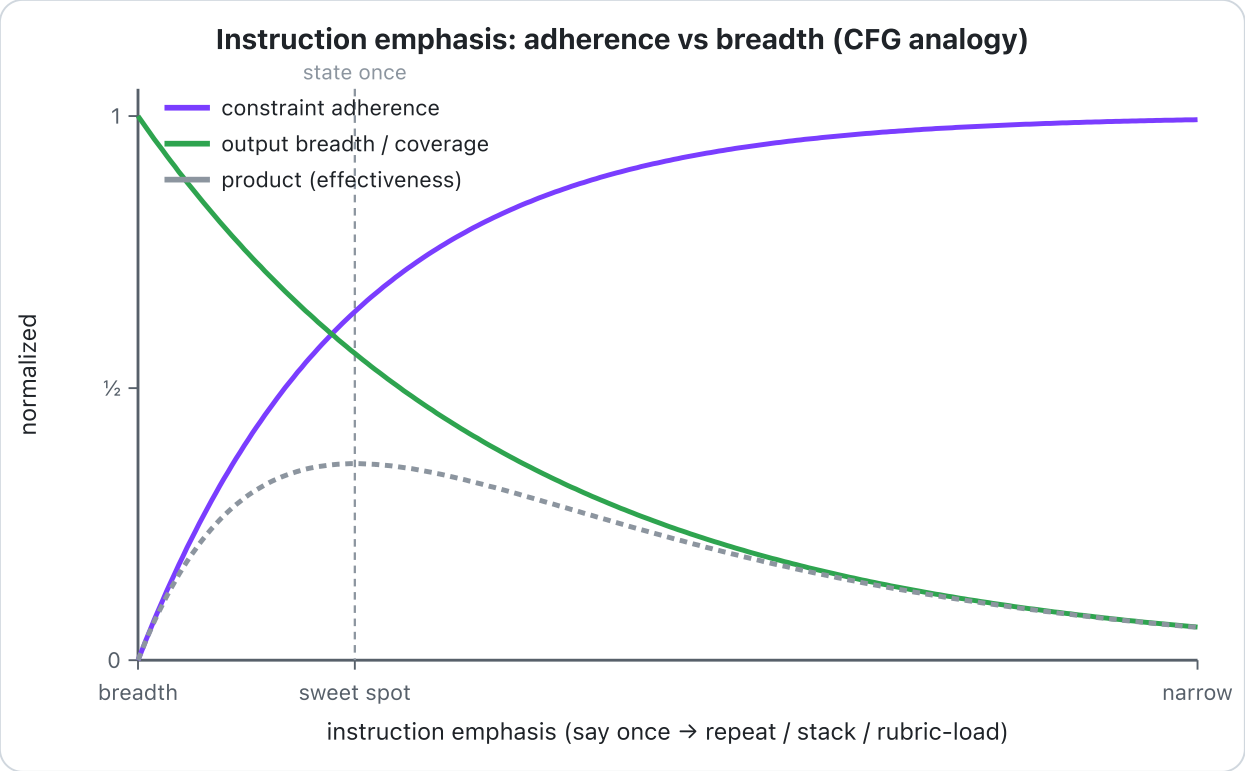

Figure 7 — CFG analogy. As instruction emphasis rises, constraint-adherence increases while output breadth falls; their product peaks at modest emphasis (R10). Classifier-free guidance measures this fidelity/diversity trade in image sampling — never about prompts or agents.

M8b Recency decay is a heuristic kernel, not a posterior (analogy)

ANALOGY

Statement. Fix a query q and a set P of curated wiki pages all tied on a primary relevance score, i.e. $\mathrm{rel}(q,p) = r > 0$ for every $p \in P$. Each page carries an age $\Delta t(p) \geq 0$ (time since last edit). Re-rank the tie by the kernel

$$w(p) \;=\; \exp\big(-\lambda\,\Delta t(p)\big)\cdot \mathrm{rel}(q,p), \qquad \lambda > 0.$$

Claim: this is a strictly monotone re-ordering in Δt — for $p_1, p_2 \in P$, $\Delta t(p_1) < \Delta t(p_2) \iff w(p_1) > w(p_2)$. Hence the induced ranking is exactly the order "most recently edited first", and ties in w occur only on ties in Δt.

Proof. On P, $\mathrm{rel}(q,\cdot) \equiv r$ is constant, so $w(p) = r\,e^{-\lambda \Delta t(p)}$ with $r > 0$. Write $w(p) = g(\Delta t(p))$ where $g(x) = r\,e^{-\lambda x}$. Then $g'(x) = -\lambda r\,e^{-\lambda x} < 0$ for all x since $\lambda, r > 0$ and $e^{-\lambda x} > 0$; thus g is strictly decreasing on $[0,\infty)$, hence injective and order-reversing. Therefore $\Delta t(p_1) < \Delta t(p_2) \Rightarrow g(\Delta t(p_1)) > g(\Delta t(p_2))$, i.e. $w(p_1) > w(p_2)$; and conversely $w(p_1) > w(p_2) \Rightarrow g(\Delta t(p_1)) > g(\Delta t(p_2)) \Rightarrow \Delta t(p_1) < \Delta t(p_2)$ by strict monotonicity. The map $p \mapsto w(p)$ thus reverses the total preorder by Δt exactly, with equal w iff equal Δt. The value of $\lambda > 0$ and of the constant r is immaterial to the order: any $\lambda > 0$ yields the identical permutation, since strict monotonicity of $e^{-\lambda x}$ holds for every positive rate. □

In fabius. archivum's M8 retrieval breaks a relevance tie on a curated wiki by preferring the more recently edited page — equivalently, it applies $w(p) = e^{-\lambda \Delta t}\mathrm{rel}$ as a deterministic recency tie-break, never as a scoring stage that can override relevance.

Honesty boundary. The exponential-decay kernel is borrowed for its shape only. The theorem above is a trivial fact about a strictly monotone scalar function; it says nothing probabilistic. In particular w is not a normalized Bayesian posterior: a genuine temporal prior would enter inside $P(\mathrm{relevant} \mid q, \Delta t, \ldots)$ and be re-normalized over the page set, whereas w is an unnormalized heuristic that merely sorts. The source — Generative Agents (Park et al. 2023) — combines recency with a fitted decay rate λ, an LLM-assigned importance score, and a reflection threshold counter over a stream of synthetic memories; those quantities are proven about that simulation, not about retrieval correctness or agent routing. fabius keeps only the monotone decay tie-break and deliberately drops the importance score, the threshold counter, and the fitted λ, so no claim of optimality or posterior calibration is made or needed.

4.7 · The operational edge — R11–R13, M9

These four rules dispatch model tiers, run long-horizon loops, compose verticals, and package the corpus. They are stated here for completeness and tagged honestly: each has a sound basis, but none has a proof folded into the coherence theorem of §5. They are the working edge, not the proven floor — and the document marks them so.

R11 Spend the cheapest model tier that holds

OPERATIONAL EDGE

Rule. Route each sub-task to the lowest-cost model tier that clears its bar, and re-tier *per sub-task*, not per session. Reserve the strong tier for ambiguity, architecture, security calls, and irreversible actions; hand mechanical, tightly-contracted work to a cheap tier. Escalate only on a *verifiable* miss (R8), never on a hunch.

Basis. This is R3's value-of-information gate lifted to model choice: among tiers t with expected loss $\mathbb{E}[L \mid t]$ and price c_t, pick $\arg\min_t(\mathbb{E}[L \mid t] + c_t)$; the strong tier earns its premium only where it strictly lowers expected loss by more than its extra cost. The escalate-on-miss mechanism is a FrugalGPT-style cascade.

Boundary. Direct on the principle (the efficiency survey's cost-effectiveness frontier); the specific savings are task- and dataset-specific and do not transfer. The gate is real; the numbers are not claimed. The benchmark's blind quality panel later measured the premise in-house: the one place the stance hurt was the small tier under the full contract on trivial one-liners — exactly the sub-tasks this rule re-tiers (§6.2).

R12 Long-horizon work runs on a loop with a dual exit gate

OPERATIONAL EDGE

Rule. For multi-cycle autonomous work, repeat the `step → verify` cycle and stop only when *both* a completion condition *and* an explicit done-signal hold. Hard-cap the cycles; on a stuck loop (no movement toward the verify condition across ~3 cycles, or a reflection that repeats the prior cause) stop and escalate — never spin. Rate-limit and checkpoint so a runaway can neither burn unbounded cost nor corrupt state.

Basis. The outer-loop form of M4's optimal-stopping rule — the same logic that caps a refine loop — applied to an autonomous build-until-done loop, with R8's convergence / no-progress test as the movement check. The dual gate is what separates "autonomous" from "infinite".

Boundary. Direct on the loop-plus-gate shape (the Ralph technique + the Reflexion stop-condition); the specific caps are operational heuristics, not derived constants.

R13 A vertical runs a studio

OPERATIONAL EDGE

Rule. A domain task spanning more than one layer (a game, a launch, a security review, a landing page) composes as a pipeline behind one goal: the **domain skill leads** (sets the WHAT and the domain laws) → `disciplina` plans and proves → the execution layers follow → `parcus` underneath. Don't collapse a vertical to a single layer, and don't let a downstream layer redefine the domain goal.

Basis. The system's own single-owner discipline (§2.2) generalized to multi-layer jobs: process picks HOW, the domain picks WHAT, and the owner of the goal is fixed for the whole pipeline.

Boundary. Direct — no external claim borrowed; it is fabius's coordination contract applied one level up.

M9 Externalize the corpus — index, not library

OPERATIONAL EDGE

Rule. Bulk reference material belongs outside the installed plugin as an indexed corpus. Each skill ships a lean entry doc plus an index and pages in only the matching slice on demand. Never bundle a new library into the artifact: it bloats the install and contradicts the lean stance the benchmark proves. Adding a capability is an index row, not a megabyte.

Basis. MemGPT paging (R9 · M7) and the memory survey's retrieval-augmented store, applied to the system's own packaging rather than to a task's working context.

Boundary. Direct on read-the-index-first. Honest current state: three full libraries still ship bundled under `references/` and are migrating behind the index — the contract is built so they can externalize without any skill changing how it reaches them.

5 · Coherence — the rules are one system

Eighteen rules drawn from eighteen papers could easily contradict one another. They do not. The capstone result of this paper is that they compose into a single decision system that is **consistent, complete, and composable**.

```
parcus  (lean floor — always on, never competes for a task verb)
  < R1   classify the 3 axes {Memory, Tools, Planning}
  < R4   scout 2–3 routes      (only if the classification is ambiguous)
  < R2   capability ladder      (admit the smallest sufficient rung)
  < R3 · M1   tool / 2nd-agent expected-value gate (confirm or veto the rung)
  < R6   plan-then-bind         (build the tool-free call DAG)
  < R5   reason → act → observe (per-edge execution invariant)
  < R7 · M2   search topology    (branch only if scorable · tree vs graph)
  < R8 · M3 · M4   refine budget  (signal-typed loop · verify depth · retry-stop)
  < M5 · M6   learning layer       (metric-gated rewrite · skill cache, queried first)
  < R9* · M7 · M8*   memory substrate (retrieve-under-budget · evict/recall · rank)
```

Theorem **The policy is one coherent decision system** CAPSTONE

Statement (Coherence Capstone). The eighteen rules $\{\text{R1--R10, M1--M8}\}$ compose into a single, well-defined decision system: a function from an incoming task to a terminating sequence of layer dispatches. We prove four properties — consistency, completeness, composability/model-applicability, and isolation of the non-governing rules.

Proof.

(a) Consistency. Strip surface vocabulary and nearly every gate is the *same object*: an expected-loss / value-of-information inequality

$$\mathbb{E}[L \mid \text{don't act}] - \mathbb{E}[L \mid \text{act}] \;>\; \text{cost},$$

instantiated on a *different* capability variable — a tool (R3: $a - b > c_{\text{call}}$), a second agent (M1: $p_a(1-p_c) > c_{\text{agent}}$), a branch (R7: $\text{EVSI} > c_{\text{search}}$), a refine loop (R8/M4), a corrector depth (M3: $qC_{\text{fail}}r'(0) > \kappa$), a rewrite (M5), a stored fact (M6/M7/R9c/M8c). A family of sign-consistent inequalities, each in its *own* variable over disjoint capability axes, is jointly satisfiable and cannot contradict itself: choosing the truth value of one variable never constrains another. Tie-breaks are uniformly toward the cheaper action (R2's $\leq$, R3/M4's strict $>$, M8c's strict $<$), so no two gates push opposite ways at a tie. The apparent "R2 minimize-machinery vs R7 branch-wide" tension dissolves on variable-counting: R2 sets the *rung-admission bit* (whether a rung enters), R7 sets the *branch-width* (how wide, once admitted) — orthogonal coordinates. Execution invariants R5 (grounding) and R6 (plan-then-bind) constrain the *order* of events, not whether any capability is added, so they live in a third, disjoint coordinate. Memory rules (M6–M8, R9–R9c) act only on the store/window variable. Disjoint variables, one sign convention $\Rightarrow$ consistent.

(b) Completeness. R1's label space $\{0,1\}^3$ is, by R1's proof, a measurable partition: every task lands in exactly one of 8 cells (mutual exclusivity + exhaustivity of ℓ^{-1}). R2's ladder is a totally ordered chain $0 \leq n \leq N$, and ρ assigns every cell a defined action, including $000 \mapsto$ parcus inline (the $n = 0$ rung). No task shape is unrouted: an empty load still routes (to parcus), and every loaded cell routes to its dominant specialist under $P \succ T \succ M$. The domain is covered with no fall-through.

(c) Composability + model-applicability. Order the firing relation

$$\text{parcus} \prec \text{R1} \prec (\text{R4 if ambiguous}) \prec \text{R2} \prec \text{R3}\cdot\text{M1} \prec \text{R6} \prec \text{R5} \prec \text{R7}\cdot\text{M2} \prec \text{R8}\cdot\text{M3}\cdot\text{M4} \prec \text{M5}\cdot\text{M6} \prec \text{R9*}\cdot\text{M7}\cdot\text{M8*}.$$

This is a strict partial order with parcus as floor and no back-edge, hence a finite acyclic DAG; a topological pass visits each rule once, so the policy *terminates*. Each rule's input is produced upstream (R1's bits feed R2; R2's admitted rung feeds R7's width; R6's schedule precedes R5's grounding check), so the composition type-checks. Every rule is stated as a closed-form predicate over quantities an LLM can estimate in-context (loads, costs, gains, hit-rates), so an LLM can execute the entire DAG from the policy text alone — model-applicability.

(d) Isolation of non-governing rules. The one qualitative rule R5 answers only *admissible / inadmissible* (its proof shows no argmax J exists) and is never traded against a score. The four analogies — R4 (flow-matching transport), R10 (classifier-free guidance), M8b (recency kernel), and M2's Graph-of-Thoughts transfer — each carry an honesty boundary confining their theorem to a non-agent domain (learned vector fields / diffusion image fidelity / a memory simulation), and each enters fabius only as a *shape* (defer-and-average, emphasis-trades-breadth, monotone tie-break, merge-needs-a-semigroup). None sits on the load-bearing inequality of (a): they decorate a decision the real-math gates have already made, and removing them changes no gate's truth value. Hence the governing skeleton is exactly the consistent inequality system of (a)–(c). □

In fabius. The capstone certifies that loading fabius installs one decision system, not eighteen competing heuristics: R1 partitions, R2–M8 each test one disjoint cost-benefit variable in a fixed acyclic order with parcus as floor, R5 guards admissibility, and the analogies advise without ever overriding a measured gate.

Scope of the theorem. Coherence is proved over the *eighteen-rule core* (R1–R10, M1–M8) — the part of the policy that rests on mathematics that governs the decision. The four operational extensions of §4.7 (R11–R13, M9) are consistent with the core in practice and reuse its objects (R11 is R3's gate over tiers; R12 is M4's optimal-stopping

rule with R8's no-progress test on the outer loop; M9 is R9/M7's paging applied to packaging), but they are **not** yet folded into this pipeline or its proof. Formalizing them — a proof per rule, then re-running the coherence check over the full set — is a deliberate next step, stated here as future work rather than an implicit claim.

6 · Does it work? The fabius benchmark

Fabius is prompt-level scaffolding, so a number measured on one model at one moment does not transfer cleanly. The defensible thing to publish is the **method, the mechanism, and the measured signal that repeats across models** — plus a one-command way to re-measure it. There is one benchmark, read out on three panels: blind-judged quality on the newest Claude models, an objective no-judge track, and blind external-model demos. In one line: **fabius improves every model it runs on — blind-judged on the newest Claude models, objectively verified by executed tests and factual checks, demoed across external families — on 20–35% less output.** Every figure below was measured on a real model at a named moment; nothing is estimated.

6.1 · The method

One test, three arms. Each task is answered three ways: `baseline` (the task only), `terse` (the task plus a generic "*be concise, write minimal code*" line), and `fabius` (the task plus the shipped stance — the shipped `AGENTS.md` plus the routed `SKILL.md`, read verbatim). The `terse` arm is the real control: beating `baseline` is easy; **beating `terse` isolates structure from mere brevity.**

The benchmark reads out on three panels. The **quality panel** is blind-judged: **two** judge models that are never told which arm produced which answer score each answer 0–5 on correctness, minimality, and best-practice, plus an objective character count; the judges disagree by only 0.72/15. The **objective panel** removes the judge entirely: generated code is *executed* against hidden test suites, and domain deliverables are graded against factual checklists by two strict graders. The **external panel** carries the same three arms cross-family through a portable harness (`evals/portable_eval.py`). A deterministic structural suite (19/19, no key, no cost) proves the skill system itself is well-formed — fifteen single-owner contracts, every reference live, the content-bound seal recomputable; the canonical receipt is `evals/results.benchmark.json`, and each panel's raw data lives in the raw receipt files under `evals/`.

6.2 · Panel A — quality, blind

Panel A — 15 tasks × 3 arms on the four newest Claude models (Fable 5 · Opus 4.8 · Sonnet 5 · Haiku 4.5); the `fabius` arm is the shipped `AGENTS.md` plus the routed `SKILL.md`, read verbatim; **two** blind judges (inter-judge gap 0.72/15), /15. Raw receipt: `evals/results.v5.json`.

Model	baseline	fabius	Δ quality	output cut
Fable	14.50	**14.73**	+0.23	−25.3%
Sonnet 5	14.07	**14.50**	+0.43	−33.7%
Opus	14.40	**14.60**	+0.20	−20.0%
Haiku	11.73	11.40	−0.33	−35.5%

The output cut is universal — 20–34% on every model. Every capable tier — Fable, Opus, and now Sonnet 5 — beats *both* controls, the bare model and the terse control; on Sonnet 5 the lift (**+0.43**) is the largest of the four tiers, and on the frontier tiers the baselines already sit near ceiling (14.4–14.5/15), so the measured effect there is same-or-better quality at 20–25% less output — not "smarter". The one miss stays on the record: Haiku dips overall. But the Haiku dip decomposes cleanly: on its twelve specialist tasks Haiku *gains* +0.71 on average (the security route 9.0→14.5, the on-chain task 10.5→14.5); the entire deficit comes from three trivial one-liners answered under the full verbatim contract (−4.50 average). A small model handed the whole stance for a task that needs none of it is exactly the failure the policy predicts — the strongest empirical case in the benchmark for model-tier routing (R11) and the `parcus` lean gate. What this panel does not show: that the stance raises frontier-model capability at ceiling;

on Fable and Opus the headroom simply is not there, though Sonnet 5's clear gain from a lower baseline shows the lift is real when there is room for it.

6.3 · Panel B — objective, no judge

Panel B — objective deliverables, judge removed: generated code *executed* against hidden test suites (4 tasks) plus domain deliverables graded against a factual checklist by two strict graders (5 tasks); 9 deliverables × 3 arms on the same four models, % of tests + checks passed. Raw receipt: `evals/results.v6.json`.

Model	baseline %	fabius %	Δ passed
Haiku	75.6	**93.0**	+17.4
Opus	84.9	**90.7**	+5.8
Sonnet 5	84.9	**90.7**	+5.8
Fable	87.2	**90.7**	+3.5

On executed algorithm code, every *bare* model already passes ~100% of the hidden tests — there is no headroom — and fabius holds the 100%. The lift lives where *looks-right* and *is-right* diverge: on the domain checklists Haiku goes 58→88, Sonnet 5 74→84, Opus 74→84, Fable 78→84. Per deliverable: the parameterized SQL route rises 67.5%→100%, the idempotent webhook 40%→67.5%, Solana account validation 47.5%→57.5%. Output fell 12–25% while pass rates rose. **Every one of the four models gains objectively** (+3.5 to +17.4). What this panel does not show: any lift on code the models already ace; and the webhook and Solana deliverables, while improved, remain below ceiling even under fabius. Improved, not solved.

6.4 · Panel C — external-model demos, and what the data supports

Panel C — external-model demos: the portable harness (`evals/portable_eval.py`) carries the same three arms cross-family, blind; 6 tasks, measured 2026-06-22 with live provider keys. The signal is the lift vs the "*be concise*" control on genuine-build, /15.

Family	lift vs terse (genuine-build, /15)
Grok	**+8.5**
GPT	**+7.0**
Claude	**+7.0**
Mistral	**+2.5**

Every family gains. Gemini is wired into the harness; consistent with the discipline of the rest of the document, no number is printed without a key.

The honest claim: **fabius gives a consistent, cross-model quality lift that grows as the model's default gets less disciplined — large at trust / order / genuine-build boundaries and where a correct-looking deliverable is not a correct one, negligible on code the models already ace.** Not "smarter." Not "10× on everything." The caveats are equally on the record: *n* is small (6–15 tasks per panel); the quality panel uses models as judges — mitigated by two blind judges that disagree by only 0.72/15 — and the objective panel removes the judge entirely, so executed hidden tests and factual checklists join the character counts as the hard signals; and every number was measured on named models at a named moment. One result moved *against* fabius and is printed anyway — Haiku's trivial-task dip on the quality panel (which decomposes into a specialist-task gain and a lean-gate miss); every other model gains on both panels.

7 · The honesty ledger

The quality bar of the whole project is one phrase: **measured, not claimed.** The sharpest decision heuristics in the policy are borrowed from a different field — the sampling mathematics of generative models. Those results are proved about numerical sampling of learned vector fields and image fidelity, *never* about agents. Where fabius uses one, it takes the shape of the idea, not a measured agent result, and says so.

DIRECT MAPPINGS

The source paper makes a claim about LLM agents that fabius applies almost verbatim: ReAct's reason→act→observe grounding contract (R5); Tree of Thoughts' branch-only-when-partials-are-judgeable gate (R7); Reflexion and Self-Refine's verifiable-signal reflection loop with diminishing returns (R8, M4); Voyager's verify-then-store skill library (M6); DSPy's contract-first metric-delta gate (M5); MemGPT's explicit paging and retrieve-over-stuff (R9, M7); and the Wang Profile/Memory/Planning/Action taxonomy anchoring the three-axis router (R1).

ANALOGY MAPPINGS

The source proves something about a different domain and fabius transfers the shape, not a measured agent result: Toolformer's loss-reduction filter is a training-time criterion against the gold continuation, uncomputable at routing time, so R3/M1 borrow only its spirit; the efficiency surveys assert diminishing returns directionally but fit no curve, so R2's knee is qualitative; Chain of Abstraction measured ~1.4× on single-tool tasks, not agent fan-out, so R6's parallel benefit is unmeasured here; Graph of Thoughts' numbers are intra-LLM, transplanted to agent topology (M2); and the three generative-model rules — Flow Matching (R4), Classifier-Free Guidance (R10), Consistency Models (M3) — together with Generative Agents' recency kernel (M8b), are **all** analogies, because those papers prove their claims about sampling and image fidelity and say nothing about agents, routing, or verification budgets.

The bar held throughout: **measured where a paper measured it in-domain; analogy everywhere the transfer crosses domains — labelled, so no claim is overstated.** Of the eighteen core rules, nine map directly from in-domain agent results and nine borrow their source only by analogy — the table of §3.2, row by row. At the level of the governing equation the ledger is stricter still: fifteen rest on mathematics that genuinely governs the decision, one (R5) is an explicit qualitative invariant, and two (R4, R10) are pure analogies — joined at the sub-rule level by M8b's recency kernel and M2's transplanted Graph-of-Thoughts numbers — that carry their caveat inline and never govern a route the way the real-math gates do.

The same discipline draws the line between the *proven core* and the *operational edge*. The coherence theorem is claimed over the eighteen-rule core only; the four extensions of §4.7 are presented as sound and useful but explicitly outside the proof. A whitepaper that quietly extended the theorem to cover features it had just added would be doing exactly the over-claiming fabius is built to refuse. The boundary is the honest thing, and it is drawn in ink, not pencil.

8 · Reproduce it, and use it

Every figure in §4 is computed, not drawn — `numpy` to SVG, no external services — and every benchmark number re-runs from one command with your own key.

```
# the figures
python3 assets/charts/render_figures.py

# the benchmark — wiring check (no key, no cost), then a real run
node evals/eval.mjs --selftest
ANTHROPIC_API_KEY=...  node evals/eval.mjs --model claude-sonnet-5
GEMINI_API_KEY=...     python evals/portable_eval.py --models gemini
```

Fabius runs as an autonomous agent on every major model, operated from the synapse console. Its brain is also the owner's working Claude Code plugin — any single `skills/<name>/` folder drops into a project's `.claude/skills/`. Because the fifteen skills are plain markdown, the portable bridge `AGENTS.md` carries the same stance to Codex, Cursor, Windsurf, Cline, Copilot, OpenCode, and Gemini — the agent operates under fabius end to end, in any tool that reads standing instructions.

The system is built to grow without bloating. Adding a capability is a new `skills/fabius-<name>/SKILL.md` with one precisely-described owned concern plus a row in the corpus index — never a megabyte bundled into the install (M9). That is how the system went from six skills to fifteen — adding the go-to-market, defensive-security, game-craft, on-chain, automation, science, AI/ML-engineering, markets, and cross-model-council verticals — while the installed brain stayed lean and the libraries moved behind the index. Each new layer links to its siblings instead of duplicating them, so the single-owner contract, and the coherence it enables, survive every addition.

9 · Conclusion

Fabius is a small idea carried all the way down. The idea is a single axis — scout wide, strike narrow — that lets an agent be thorough and minimal at once because the two live on different axes. The system is fifteen coordinated skills with a single-owner contract. The policy is eighteen proven rules — each sourced to a real result — and four operational extensions held honestly at the edge. And the mathematics is not decoration: every gate is the same value-of-information threshold applied to a different capability, composed in an acyclic pipeline that routes every task and terminates — proved here, and adversarially checked.

What makes it trustworthy is the line it refuses to cross. Where a paper measured a thing in its own domain, fabius uses the result; where the transfer would cross domains, it borrows only the shape and labels it. The benchmark says the same thing in data: a consistent, cross-model lift that grows as the model's default discipline drops — shorter answers a blind judge scores higher, and, when the judge is removed, more hidden tests and factual checks objectively passed — and nothing more than that. **Measured, not claimed.**

Scout wide, strike narrow.

References

Agent research

Yao, S. et al. *ReAct: Synergizing Reasoning and Acting in Language Models.* 2022.

Schick, T. et al. *Toolformer: Language Models Can Teach Themselves to Use Tools.* 2023.

Shinn, N. et al. *Reflexion: Language Agents with Verbal Reinforcement Learning.* 2023.

Madaan, A. et al. *Self-Refine: Iterative Refinement with Self-Feedback.* 2023.

Yao, S. et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models.* 2023.

Besta, M. et al. *Graph of Thoughts: Solving Elaborate Problems with Large Language Models.* 2023.

Gao, S. et al. *Efficient Tool Use with Chain-of-Abstraction Reasoning.* 2024.

Packer, C. et al. *MemGPT: Towards LLMs as Operating Systems.* 2023.

Park, J. S. et al. *Generative Agents: Interactive Simulacra of Human Behavior.* 2023.

Wang, L. et al. *A Survey on Large Language Model based Autonomous Agents.* 2023.

Khattab, O. et al. *DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines.* 2023.

Wang, G. et al. *Voyager: An Open-Ended Embodied Agent with Large Language Models.* 2023.

Lipman, Y. et al. *Flow Matching for Generative Modeling.* 2023.

Ho, J. & Salimans, T. *Classifier-Free Diffusion Guidance.* 2022.

Song, Y. et al. *Consistency Models.* 2023.

Hao, S. et al. *Reasoning with Language Model is Planning with World Model (RAP).* 2023.

Toward Efficient Agents — efficiency / cost-effectiveness survey. arXiv:2601.14192, 2026.

A Survey on Memory for LLM Agents. arXiv:2603.07670, 2026.

Mathematical foundations

Howard, R. A. *Information Value Theory.* IEEE Trans. SSC, 1966.

Raiffa, H. & Schlaifer, R. *Applied Statistical Decision Theory.* 1961.

Banach, S. *Sur les opérations dans les ensembles abstraits.* Fund. Math., 1922.

Nemhauser, G., Wolsey, L. & Fisher, M. *An analysis of approximations for maximizing submodular set functions.* 1978.

Khuller, S., Moss, A. & Naor, J. *The budgeted maximum coverage problem.* 1999. · Sviridenko, M. 2004.

Kraft, L. G. 1949 · McMillan, B. *Two inequalities implied by unique decipherability.* 1956.

Shannon, C. E. *Coding theorems for a discrete source with a fidelity criterion.* 1959.

Robertson, S. E. *The Probability Ranking Principle in IR.* 1977.

Vapnik, V. *The Nature of Statistical Learning Theory.* 1995. · Hoeffding, W. 1963.

Bellman, R. *Dynamic Programming.* 1957. · Michie, D. *"Memo" functions.* 1968.

Rissanen, J. *Modeling by shortest data description (MDL).* 1978.

Brent, R. P. *The parallel evaluation of general arithmetic expressions.* 1974.