The Operating Architecture of the AI-Native Business: AGC Before AGI

Alan McCord, Ph.D.

mccord.alan@gmail.com

Pre-note on GBrain and scope. The AGC/*Athanor* program was developed independently and predates the public GBrain company-brain framing. GBrain is an important memory and graph-synthesis layer for agents; the present work addresses a broader operating problem: business cognition with deep combinatorial complexity, where intents, skills, harnesses, permissions, costs, latency, validation, compilation, and evolutionary improvement is jointly governed.

This is the second of three papers on the operating architecture of the AI-native business. *Paper I, Composition Beats Bundling*, showed why late-bound skill-to-harness assignment can dominate fixed agent bundles. The present paper introduces **Artificial Generalizing Competence (AGC)** as the business-relevant replacement target for AGI. *Paper III* defines in detail how the knowledge system is built and maintained. Together the trilogy marks the death of human-designed agents as the central business artifact: the future company brain assigns, compiles, validates, prices, secures, and improves work under formal gates. A system instantiating these principles is *Athanor*: an investable operating model for the AI-native enterprise, where every successful capability is captured, compiled, and compounded into governed institutional intelligence instead of disappearing into another fragile one-off agent.

Abstract

Paper I formalizes why assigning top-k skills to a general harness can dominate routing intents to statically pre-bundled agents under explicit capacity, retrieval, and distractor-cost assumptions. This paper addresses *how to operate* that architecture when those assumptions must be measured rather than presumed: which harness topology to run, how to know whether an answer can be trusted, at what accuracy/latency/cost operating point to run, and how the system should improve over time. A single discipline underlies the operation — the intent is *resolved early*, once, at the input boundary, while the skills are *bound late*, at the harness — placing each commitment where it is best made. We give one main theorem and four supporting propositions. **Proposition C (harness selection):** the preferred harness topology can be read from structural features of the *retrieved skill subgraph* — its parallel width, dependency depth, verifiable nodes, and risk annotations — when the loss model is structurally sufficient. **Theorem D (selective reliability):** when intent space is partitioned by an explicit, per-node-validated taxonomy, the system can route every intent to {answer, warn, abstain} with a coverage–risk guarantee — the conditional failure rate on the accepted set is provably bounded, with simultaneous confidence across accepted leaves — and this guarantee is *architectural*: a system without a validated partition cannot construct it and fails silently off-distribution. **Proposition D-P (authorization soundness):** the permission graph is a separate safety invariant: every emitted answer can use only authorized skills, tools, and data, and unauthorized intents cannot be answered even when the model would otherwise comply. **Proposition E (operating-point**

frontier): a composition system spans a dense accuracy/latency/cost Pareto frontier from a linear set of primitives; a user declares an operating constraint and the system selects the accuracy-maximal feasible point; and abstention becomes *budget-dependent*. **Proposition F (joint configuration selection):** selecting topology, model, and budget together is a constant-size, regret-bounded grid decision; it decomposes into Propositions C and E when topology preference is model-invariant. We then show the abstention log is the demand signal for a closed skill-evolution loop that expands the answerable frontier and — because each skill is bound to a fixed set of intents — also optimizes existing skills by automated search against a stationary fitness signal, subject to validation and regression gates. We name the resulting engineering target **Artificial Generalizing Competence (AGC)**, and define it precisely (§8) as competence that is *scoped* to a validated intent set, *bounded* by an explicit declared frontier, *priced* per intent, and *monotonically extended* over time — every clause a quantity a business can audit and contract, in pointed contrast to the unmeasurable predicates of Artificial General Intelligence. The result is a formal operating architecture for company brains: self-improvement becomes a gated theorem, economics becomes a selectable frontier, and security becomes an enforced graph invariant. Its signature behavior is the calibrated "I cannot reliably answer that yet": the honest report of a frontier, never a claim of completion.

1 Introduction

This is the *Attention Is All You Need* moment for company brains: the claim is not that one more giant model will become a business operating system, but that a company brain must be an auditable competence engine. For business deployment, AGI is the wrong target: as an operating specification, it is completely irrelevant. A company does not need an oracle that claims general intelligence. It needs a system that knows which work it can perform, which work it cannot yet perform, which data it may touch, how much reliability costs, how fast an answer can be obtained, and how its competence expands without breaking what already works. This paper formalizes that target as **Artificial Generalizing Competence (AGC)**. AGC makes self-learning a validation-gated skill-evolution loop; makes economics an explicit accuracy/speed/cost frontier; and makes security an authorization theorem over a permission graph. It states what the future *Company Brain* must become: not a chatbot, not an agent bundle, and not an AGI slogan, but a priced, permissioned, auditable, self-extending decision system. Paper III is the builder manual for that system: it explains how to construct and maintain the ontology, intent taxonomy, and skill graph on which the present paper's guarantees depend.

Paper I established a conditional separation: late-bound skill-to-harness assignment — a retriever delivering the top-k skills of a shared library to a single general harness — improves capacity relative to static fixed agents and can improve robustness when retrieval sharpness and distractor cost fall in the favorable regime. Its latest live evidence makes the retrieval premise operational rather than decorative: unstructured hybrid retrieval is a high-library boundary case, while no-ID semantic facets preserve rank concentration through `N_S=1600`. A natural reading is that the architecture question is closed. It is not. An architecture that *can* express the right solution still has to be *operated*: a control-flow topology must be chosen, a base model and a compute budget committed, an answer either trusted or withheld, and — over time — the system's competence extended to demand it cannot yet serve.

These operational questions are exactly where fixed-agent systems fail quietly. A fixed agent is built at one control-flow topology, one model, one cost point; it has no representation of where it has and has not been validated; and improving it means hand-authoring another agent. This paper shows that skill-to-harness assignment, *because* it composes at inference time, turns each of these into a

well-posed decision problem under explicit validation assumptions, with a guarantee attached where those assumptions hold.

This is also the end of the hand-coded agent as the primary unit of enterprise AI. Humans still define meanings, policies, validation standards, and high-value skills, but they no longer freeze those choices into one-off agents. The system assigns skills to harnesses at run time, compiles mature skills into deterministic or solver-backed implementations when evidence justifies it, and admits improvements only through validation and regression gates. The result is a formal framework for self-improvement rather than a pile of edited prompts. DeepMind showed the power of automated improvement in closed game worlds through self-play (Silver et al., 2017); AGC moves that pattern into business operations, replacing game score with validated success, permission soundness, latency, cost, and regression preservation.

Resolve early, bind late. A pipeline from a user to an outcome contains two binding decisions — *which intent* the user means, and *which skills* serve it — and they belong at opposite ends. The intent is bound **early**: resolved once, at the input boundary, where the user is present to clarify and a shared domain ontology normalizes terminology (§3). The skills are bound **late**: kept a soft, ranked set by the retriever and committed only at the harness, the component most able to judge the task — Paper I's *late-binding routing*. The rule is the same in both directions: make each commitment where it can be made best, and make it once. Everything in between — topology, model, budget, the abstention decision — then operates on a settled intent and a settled skill set, which is exactly what makes the decision problems of §§4–6 well-posed.

The shift in framing. We argue the field's nominal target — *intelligence* — is the wrong object for deployment. Intelligence is unmeasurable and unfalsifiable. What a deployed system owes its operator is *competence*: success on a specified task, at a known reliability, latency, and cost. Competence is measurable precisely because it is defined relative to a task and a standard — and the validation sets that skill-to-harness assignment already requires *are* that standard. We call the target **Artificial Generalizing Competence (AGC)**. The participle is deliberate: *generalizing*, not *general*. A system should extend its competence; it should never claim to have finished. AGC's signature behavior is the calibrated refusal — "I cannot reliably answer that yet" — which is not a limitation bolted on but the explicit, honest report of the system's current frontier. It also inverts the usual deployment recipe of building a general model and then adding guardrails: in AGC, governance is structural, not retrofitted (§8).

Contributions.

1. A model (§3) of the *domain ontology*, the *retrieved skill subgraph*, the *intent taxonomy*, the *input-boundary resolution stage*, the *harness configuration*, and the *operating point*, recapping and extending Paper I's setup.
2. **Proposition C** (§4): harness topology selection is a structural readout of the retrieved skill subgraph under an explicit structural-loss assumption; the readout rule dominates any fixed topology only to the extent that assumption is valid.
3. **Theorem D** (§5): a coverage–risk guarantee for calibrated abstention, available only to systems built on an explicit, validated intent taxonomy. We connect it to Mondrian conformal prediction.
4. **Proposition D-P** (§5): authorization soundness from the permission graph — every emitted answer is restricted to skills, tools, and data authorized for the user/context, independently of model compliance.
5. **Propositions E and F** (§6): the accuracy/latency/cost operating point is a composed

object spanning a dense Pareto frontier; the user selects a feasible point; abstention is budget-dependent, yielding a per-intent reliability–cost curve; and joint selection of topology, model, and budget is a constant-size, regret-bounded optimization that decomposes exactly into Propositions C and E under a model-invariance condition.

6. The **skill-evolution loop** (§7): the abstention log as a demand signal for *extensive* frontier expansion, and the fixed skill–intent linkage as a stationary fitness signal for *intensive*, automated per-skill optimization.
7. A precise **definition of AGC** and the thesis around it (§8): AGC is the conjunction of the paper's guarantees — competence that is scoped, boundary-declared, priced, permissioned, and monotonically extended — a target that is constructible and checkable, and deliberately not AGI.

Throughout, §4, §6, and Paper I's Theorem A are shown to share the same design pattern — the *Composition Separation Principle* — on three different axes.

2 Related Work

Selective prediction and abstention. The reject option for classifiers is classical (Geifman & El-Yaniv, 2017); the goal is to bound error on the accepted set while maximizing coverage. For LLMs, abstention is now an active subfield with dedicated benchmarks; recent evaluation finds that even strong reasoning models abstain poorly on unanswerable or underspecified questions (Kirichenko et al., 2025), and surveys catalogue the spectrum of refusal behaviors (Wen et al., 2024). Most of this work attempts to make the *model* introspect its own uncertainty. Theorem D takes the opposite stance: abstention is grounded in an *external, explicit* validated partition of intent space, not in model introspection — which is what converts it from a heuristic into a guarantee. The statistical tool is conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2023); the taxonomy leaves are the natural categories of a *Mondrian* conformal predictor, which delivers category-conditional validity.

Cost-aware routing and inference scaling. A line of work routes queries between models of differing cost and quality — RouteLLM (Ong et al., 2024), FrugalGPT (Chen et al., 2023), RouterBench (Hu et al., 2024) — and compute-optimal test-time scaling allocates inference compute per prompt by difficulty (Snell et al., 2024). These select among a small set of pre-specified options or scale a single knob. Proposition E generalizes the idea: the operating point is a *multi-dimensional composed configuration* (model $\times$ topology $\times$ k $\times$ iteration budget), it is *user-declared* rather than only difficulty-inferred, and it is coupled to the abstention guarantee of Theorem D.

Harness topologies. The control-flow structures we select among are established individually: ReAct interleaves reasoning and acting (Yao et al., 2023); Reflexion adds a verbal self-critique loop (Shinn et al., 2023); multi-agent debate aggregates contested answers (Du et al., 2023); manager/worker decomposition handles parallel subtasks. Each is typically deployed as a fixed choice. Proposition C makes the choice a function of task structure under an explicit loss model.

Skill libraries and lifelong learning. Voyager (Wang et al., 2023) demonstrated an ever-growing library of composable code skills with retrieval and an iterative refinement loop, driven by an *exploration* curriculum. The skill-evolution loop of §7 differs in its driver: it is *demand*-driven — the abstention log of real failed intents — which aligns growth with economic value and makes "what to learn next" an observable rather than a heuristic.

3 Model

We adopt the model of Paper I (§3–4) and extend it for the governance layer; shared symbols carry identical meaning. The core notation is summarized below; theorem-specific symbols ($\phi, \varepsilon, \alpha, \mathcal{A}_\varepsilon, \mathcal{G}, \ldots$) are defined where introduced.

Symbol	Meaning
$\mathcal{S},\ N_S$	skill library and its size (Paper I)
$\mathcal{I},\ S^\star(i),\ r$	intents; required skill set of i; arity (Paper I)
k	number of parent skills retrieved per intent (Paper I)
$G(i)$	retrieved skill subgraph for intent i
e_s	interpreted-execution error rate of skill s
$\mathcal{T},\ c$	intent taxonomy; a leaf of it
$V_c,\ n_c,\ \hat{q}_c$	leaf c's validation set, its size, its empirical success rate
$\kappa(i)$	classifier confidence that i lies in leaf $c(i)$
$\mathcal{H},\ h$	set of harness topologies; a harness configuration
$(a, \ell, \gamma)(h, i)$	operating point of h on i: accuracy, latency, cost

Domain ontology. Beneath the skill and intent structures sits a *domain ontology*: a knowledge graph of the entities, metrics, KPIs, and terminology of the deployment domain, together with their canonical definitions and relations. Skills, intents, and validation criteria are all expressed against it. The ontology is *injected into the harness context*, so that every base LLM, every harness topology, and every skill definition resolves a term — "net revenue retention", "active account", "fiscal Q3" — to one and the same referent. This shared grounding is what makes the skill graph a graph of *commensurable* nodes rather than a bag of independently authored tools; it is also the semantic invariant that lets Propositions C, E, and F swap topologies, models, and budgets coherently — the machinery underneath can change only because the meaning of what it operates on is held fixed.

Skill graph. The library $\mathcal{S}$ (N_S skills) is a directed graph. Edges are typed: `requires` (s_1 must run before s_2), `verifies` (s_1 can check s_2's output), `refines`, `conflicts-with`. Each skill carries annotations: an interpreted-execution error rate e_s and a stakes/risk weight. Skills may additionally be *compiled* — replaced by deterministic code or an exact solver where their problem type permits (Paper I, §9) — in which case their operating point is near-deterministic, a fact the cost estimates of §6 inherit. The skill graph is one of Athanor's knowledge structures, indexed — like the intent taxonomy and permission graph below — against the domain ontology.

Retrieved skill subgraph. For intent i, retrieval returns the top-k parent skills $T_k(i)$; the *retrieved subgraph* $G(i)$ is the subgraph of the skill graph induced by $T_k(i)$ — those skills and the typed edges among them, with their annotations.

Hierarchical skills and progressive disclosure. As in Paper I (§4), the library is subtree-structured: each entry is a *parent skill* with a short descriptor, beneath which fuller instructions and sub-skills are nested. Retrieval returns the top-k *parent* skills; the harness then performs *progressive disclosure*, expanding a parent into its `SKILL.md` body and nested sub-skills only when it engages that parent (Anthropic, 2025). The retrieved subgraph $G(i)$ therefore carries this subtree structure, and progressive disclosure is itself a within-skill control-flow step available to the feature map of §4.

Intent taxonomy. Deployment intent space is partitioned by a taxonomy $\mathcal{T}$ into leaves $\{c\}$. Each leaf c has a validation set V_c of n_c intents with known correct outcomes. A classifier maps an

incoming intent i to a leaf $c(i)$ with confidence $\kappa(i)$.

Input boundary and intent resolution. A deployment does not receive clean intents; it receives raw user input — utterances, session chatter, under-specified or variably phrased requests. Athanor resolves this at a single *input boundary*: a resolution stage $\mathcal{R}$ maps raw input to a canonical intent i in the taxonomy, normalizing synonyms and variant phrasings against the domain ontology and, where the input is ambiguous, asking a clarifying question before committing. Everything downstream — retrieval, harness selection, the {answer, warn, abstain} routing of §5 — consumes the resolved canonical intent. This is the input-side counterpart of Paper I's *late binding*: skills are bound late, at the most capable component, while the intent is resolved early and once, at the boundary, so that ambiguity is never re-litigated by every harness and skill downstream.

Harness and operating point. A *harness topology* is drawn from a small set $\mathcal{H}$ (single-shot, plan–execute, manager/worker, actor–critic, debate, ...); $|\mathcal{H}|$ is a handful. A *harness configuration* is $h = (\text{topology},\ \text{LLM},\ k,\ \text{iteration budget},\ \text{tool set})$. Each h induces, for intent i, an *operating point* $(a(h,i),\ \ell(h,i),\ \gamma(h,i))$ — expected accuracy, latency, and compute cost.

Permission graph. Orthogonally, a permission graph specifies which skills, tools, and data a given user or context may invoke. For user or context u, write $\text{Perm}_u(i) = 1$ when every skill, tool, and datum required by intent i is authorized for u. The executor must enforce the graph, so unauthorized resources are not merely discouraged but unavailable. The graph enters §5 as a second gate: abstention covers not only "untested" but also "not permitted".

4 Proposition C — Harness Selection as a Structural Readout

The control-flow topology should match the *computational shape* of the task. That shape is carried by the retrieved subgraph: mutually independent skills admit parallel fan-out; `requires` chains force sequential execution; a high-error skill with an incoming `verifies` edge wants a critic loop; a single low-stakes skill wants one shot. We formalize this.

Structural feature map. For $G(i)$ define $\phi(G(i)) = (w, d, \mathcal{V}, \{e_s\}, \{\text{stakes}_s\})$, where w is the parallel width (largest set of mutually `requires`-independent skills), d the dependency depth (longest `requires` path), $\mathcal{V}$ the set of skills with an incoming `verifies` edge, e_s the per-skill error rate, stakes_s the per-skill risk weight.

Loss model (C1). Executing i under topology h incurs expected loss $\mathcal{L}(h,i) = \beta_{\text{err}} \cdot \text{Err}(h,i) + \beta_{\text{lat}} \cdot \text{Lat}(h,i)$, where the error and latency of each topology are determined by $\phi(G(i))$: error compounds over skills on the executed path and is reduced on verified skills by a critic loop; latency tracks the critical-path length, which a parallel topology shortens from total count to depth d. (C1) is the assumption that Err and Lat are functions of $\phi(G(i))$ alone.

Proposition C. *Under (C1):*

(1) Structural sufficiency. $\mathcal{L}(h,i)$ depends on i only through $\phi(G(i))$. Hence the optimal topology $h^\star(i) = \arg\min_{h \in \mathcal{H}} \mathcal{L}(h,i)$ is a function of $\phi(G(i))$ and the loss weights alone; the intent label carries no further information.

(2) Dominance. The rule $i \mapsto h^\star(\phi(G(i)))$ weakly dominates every fixed topology $h_0 \in \mathcal{H}$ in expected loss, and strictly dominates it whenever the distribution of $\phi(G(i))$ places positive mass on two structural regimes with distinct minimizers.

(3) No blowup. The decision rule has output cardinality $|\mathcal{H}|$, independent of N_S and of the number of intents M.

Proof. (1) is immediate from (C1): if Err and Lat are functions of $\phi(G(i))$, so is $\mathcal{L}(\cdot, i)$, and so is its argmin over the finite set $\mathcal{H}$. (2) For each i, $\mathcal{L}(h^\star(i), i) \le \mathcal{L}(h_0, i)$ by definition of the argmin; taking expectations over i gives weak dominance. If two regimes R_1, R_2 with $\Pr[R_1], \Pr[R_2] > 0$ have arg min topologies $h_1 \neq h_2$, then no single h_0 minimizes on both, so on at least one regime h_0 is strictly worse, making the inequality strict in expectation. (3) The rule's codomain is $\mathcal{H}$. ■

Remarks.

This is the Composition Separation Principle on the control-flow axis. A small set of composable topologies covers a combinatorially large space of retrieved subgraphs, exactly as a small skill library covers a combinatorially large intent space (Paper I, Theorem A). A system that instead *pre-assigns* a topology per intent maintains $\Theta(M)$ assignments; the structural readout maintains one rule of output cardinality $|\mathcal{H}|$.

The load-bearing assumption is (C1). Topology loss is not purely structural — base-LLM capability and operating constraints also matter. Those enter not here but in Proposition E, which selects model and budget; Proposition C selects topology given them. The honest scope of C is: *given* the operating point, topology is a structural readout.

Validation supplies the features. The per-skill error rates $\{e_s\}$ are not free parameters; they are measured on validation data — the same asset that Theorem D consumes. Mis-estimating w, d, or $\{e_s\}$ is the failure mode of C, and it is bounded by validation quality.

Progressive disclosure is within-skill control flow. Each retrieved parent skill is a subtree (§3); the harness's decision to expand it — to disclose its `SKILL.md` body and sub-skills — or to leave it as a descriptor stub is a small control-flow choice nested inside the topology. Subtree depth and branching are accordingly structural features the map ϕ can read, and an unexpanded parent contributes neither error nor latency — the mechanism, shared with Paper I's Theorem B, by which a generous retrieval count k stays cheap.

5 Theorem D — Selective Reliability

A deployed agent owes its operator not just answers but a calibrated account of which answers can be trusted. We show that an explicit, validated intent taxonomy makes "I cannot reliably answer that yet" a *guarantee* rather than a hope.

Assumptions. (D1) *Intra-leaf exchangeability:* within each taxonomy leaf c, validation intents and deployment intents are exchangeable. (D2) *Classification soundness on the confident region:* on the set $\{i : \kappa(i) \ge \tau\}$, the taxonomy classifier assigns i to a leaf containing it. (D3) *Permission soundness and enforcement:* $\mathrm{Perm}_u(i) = 1$ only when every required skill, tool, and datum for intent i is authorized for user or context u, and the executor cannot invoke unauthorized resources.

Construction. Fix a risk tolerance ε and confidence $1-\alpha$. Assign per-leaf confidence budgets $\{\alpha_c\}$ with $\sum_c \alpha_c \le \alpha$. For leaf c with n_c validation intents and empirical success $\hat{q}_c$, let $\mathrm{LCB}_{\alpha_c}(\hat{q}_c, n_c)$ be a lower confidence bound on the true leaf success rate (Clopper–Pearson, Hoeffding, or a Mondrian conformal bound). For user or context u, define the **acceptance region**

$$\mathcal{A}_\varepsilon(u) \;=\; \{\, i \;:\; \kappa(i) \ge \tau,\ \mathrm{Perm}_u(i) = 1, \;\text{ and }\; \mathrm{LCB}_{\alpha_{c(i)}}\big(\hat{q}_{c(i)},\, n_{c(i)}\big) \;\ge\; 1 - \varepsilon \,\}.$$

Operationally, the router emits **answer** when $i \in \mathcal{A}_\varepsilon(u)$; **warn** when $\kappa(i) \geq \tau$ and $\text{Perm}_u(i) = 1$ but the validation lower bound is below threshold; and **abstain** when $\kappa(i) < \tau$, the intent is out of taxonomy, or $\text{Perm}_u(i) = 0$.

Theorem D. *Under (D1)–(D3), the deployment failure rate conditional on acceptance satisfies*

$$\Pr\left[\,\text{fail} \;\middle|\; i \in \mathcal{A}_\varepsilon(u)\,\right] \;\leq\; \varepsilon$$

with probability at least $1-\alpha$ over the draw of validation sets, simultaneously across accepted leaves. Intents with $i \notin \mathcal{A}_\varepsilon(u)$ are routed to warn *when they are confidently placed and permitted but insufficiently validated, or* abstain *when they are out of taxonomy, ambiguous after boundary resolution, or not permitted. The abstention/warn mass equals the deployment mass on regions that are insufficiently validated, out of taxonomy, unresolved, or unauthorized.*

Corollary (architectural necessity). *A system without an explicit, validated partition of intent space cannot construct $\mathcal{A}_\varepsilon$: it has no per-region success estimate $\hat{q}_c$, hence no acceptance test, and therefore cannot bound its conditional failure rate. It fails silently off-distribution. The coverage–risk guarantee is available only to architectures built on a validated intent taxonomy.*

Proof. For each leaf c, the lower confidence bound gives $q_c \geq 1-\varepsilon$ with probability at least $1-\alpha_c$ whenever $\text{LCB}_{\alpha_c}(\hat{q}_c, n_c) \geq 1-\varepsilon$. A union bound over leaves gives simultaneous validity with probability at least $1-\sum_c \alpha_c \geq 1-\alpha$. Take an accepted intent i, so $\kappa(i) \geq \tau$, $\text{Perm}_u(i) = 1$, and the lower bound for $c(i)$ is above threshold. By (D2), i lies in leaf $c(i)$. By (D1), that leaf's validation success rate is an exchangeable estimate of its deployment success rate, and by the simultaneous event just established, $q_{c(i)} \geq 1-\varepsilon$. Deployment failure on accepted intents is therefore $\leq \varepsilon$ leafwise and hence after aggregation over accepted leaves. The corollary follows because every step consumed $\hat{q}_c$, which an unpartitioned system does not possess. ■

Proposition D-P (authorization soundness). *Under (D3), every emitted* answer *uses only skills, tools, and data authorized for user or context u. If an intent requires an unauthorized resource, then $\text{Perm}_u(i) = 0$, so $i \notin \mathcal{A}_\varepsilon(u)$ and the router cannot emit* answer *for that intent. Equivalently, the permission graph is an execution-time safety invariant: model willingness to answer cannot bypass an unavailable skill, tool, or datum.*

Proof. The acceptance definition includes $\text{Perm}_u(i) = 1$ as a necessary condition. By (D3), that condition is sound and enforced at execution time, so an accepted run cannot invoke unauthorized resources. If a required resource is unauthorized, the predicate is false and the intent is outside the answer region. ■

Remarks.

Why this differs from model-introspective abstention. Most LLM abstention methods ask the model to judge its own uncertainty, and benchmarks show this is unreliable (Kirichenko et al., 2025). Theorem D does not ask the model anything: the acceptance test is computed from an external structure — the taxonomy and its validation record. The guarantee is a property of the *architecture*, transferable across base models.

Mondrian conformal prediction. Taking LCB_{α_c} from a conformal procedure run *within each leaf* makes the bound distribution-free under exchangeability; the taxonomy leaves are precisely the categories of a Mondrian conformal predictor, which is designed to deliver category-conditional

validity (Vovk et al., 2005). The per-leaf α_c allocation is the simultaneous-confidence bookkeeping that prevents the guarantee from silently degrading as the taxonomy grows.

The granularity tradeoff. (D1) is load-bearing. A coarse leaf bundles heterogeneous intents and weakens exchangeability; a fine leaf restores it but costs more validation. Taxonomy granularity is therefore a tunable: finer taxonomy means a stronger guarantee and higher validation cost. This is the price of the guarantee, and it should be reported, not hidden.

Two failure modes, two gates. Acceptance is a conjunction. The $\kappa \geq \tau$ gate guards against *misplacement* (an out-of-taxonomy intent scored against a leaf it does not belong to); the LCB gate guards against *under-validation* (a correctly placed intent in a leaf with too little or too weak validation). Both are necessary; dropping either voids the guarantee.

Ambiguity is resolved at the boundary, not downstream. The {answer, warn, abstain} routing is preceded by the input-boundary resolution stage $\mathcal{R}$ (§3), and this sharpens the $\kappa \geq \tau$ gate. Low classifier confidence has two causes that demand opposite responses: input that is *under-specified* — genuine ambiguity — should not be abstained on but *clarified*, and a successful clarification raises κ and re-enters the routing; input that is genuinely *out of taxonomy* should abstain. The boundary stage separates these against the ontology and the taxonomy, once, at the edge. Clarification is therefore a boundary-side recourse that converts a class of would-be abstentions into answered intents without weakening the Theorem D guarantee — because it raises κ *before* the acceptance test rather than bypassing it. It is the input-side analogue of Paper I's slack parameter, which converts a class of would-be misroutes into successes.

The permission gate. The same {answer, warn, abstain} machinery absorbs the permission graph: an intent requiring a skill or datum the user may not access is routed to abstain on *authorization* grounds. Athanor thus presents a single, unified "cannot or should not" boundary — one mechanism for *untested* and *not permitted*. Proposition D-P is intentionally narrower than Theorem D: it proves authorization safety, not predictive reliability or policy correctness. It says that once the policy graph is specified and enforced, answers cannot cross it.

Staleness. Validation estimates age as the world drifts; (D1) erodes over time. This is not a defect to be patched but the reason §7's evolution loop must also *re-validate*: the guarantee is maintained, not established once.

6 Propositions E–F — The Operating Point and Joint Configuration Selection

A business user does not want a fixed accuracy; they want to choose, per session, a point in the accuracy/latency/cost trade-off. We show skill routing makes this a controllable decision and that fixed agents in the model cannot do so without enumerating comparable configurations (Proposition E), and that the full configuration can be selected jointly with bounded regret (Proposition F).

Configurations. Let the system maintain L base LLMs, $|\mathcal{H}|$ topologies, K admissible values of the retrieval count k, and I iteration budgets. A configuration h picks one of each; it induces an operating point $(a(h,i), \ell(h,i), \gamma(h,i))$.

Proposition E.

(1) Separation. The composition system reaches $\Theta(L\,|\mathcal{H}|\,K\,I)$ distinct operating configurations from $O(L + |\mathcal{H}| + K + I)$ maintained primitives. A fixed-agent system realizes

one operating point per pre-built agent; to populate a frontier of comparable density it must maintain $\Omega(L\,|\mathcal{H}|\,K\,I)$ agents.

(2) Constrained selection. Given a user operating constraint — a cost ceiling B, a latency ceiling $L_{\max}$, or a scalarization of the three objectives — the system returns

$$h^\star(i) \;=\; \arg\max_h \; \{\, a(h,i) \;:\; \gamma(h,i) \le B,\; \ell(h,i) \le L_{\max} \,\},$$

the accuracy-maximal feasible point on the per-intent frontier. A fixed agent is a single point: feasible or not, with no interior recourse.

(3) Budget-dependent abstention. Let $\mathcal{A}_\varepsilon(B,u)$ be the acceptance region of Theorem D for user or context u, recomputed with the accuracy $a(h^\star(i),i)$ achieved at the budget-optimal configuration. Then $\mathcal{A}_\varepsilon(B,u)$ is monotone non-decreasing in B, and if accuracy is strictly budget-sensitive on a set of permitted intents of positive mass, then for $B' > B$ there exist intents in $\mathcal{A}_\varepsilon(B',u) \setminus \mathcal{A}_\varepsilon(B,u)$.

Proof. (1) The configuration count is the product of the knob cardinalities; the maintained primitives are their sum. A fixed agent embodies exactly one configuration, so matching the product requires that many agents. (2) is the definition of the constrained maximizer over the reachable set; a fixed agent's reachable set is a singleton. (3) Raising B enlarges the feasible set $\{h : \gamma(h,i) \le B\}$, so the constrained maximum $a(h^\star(i),i)$ is non-decreasing in B; hence the acceptance test, monotone in accuracy for a fixed permission context u, can only gain permitted intents — $\mathcal{A}_\varepsilon(B,u) \subseteq \mathcal{A}_\varepsilon(B',u)$. Strictness: if some positive-mass permitted intent's optimal accuracy strictly increases from below $1-\varepsilon$ to at or above it as the budget rises from B to B', that intent enters $\mathcal{A}_\varepsilon(B',u) \setminus \mathcal{A}_\varepsilon(B,u)$. ■

Corollary (session-level allocation). *Given a session of T intents and a session budget B_{sess}, allocating the budget to maximize total expected success $\sum_t a(h_t,i_t)$ subject to $\sum_t \gamma(h_t,i_t) \le B_{sess}$ is a bounded-knapsack problem; its solution funds structurally hard intents and starves easy ones. A fixed-agent system cannot reallocate within a session, since each agent's cost is fixed.*

Remarks.

The genuinely new object is the per-intent reliability–cost curve. Parts (2) and (3) together let the system tell a user, for a specific intent: *"At your declared budget this intent is in the warn region; raising the budget to B' moves it to answer."* The user is not handed a fixed reliability — they are handed a curve and choose a point on it. This is the unification of Paper I's robustness result with Proposition C and Theorem D into a single decision object: capability, control flow, reliability, and cost are jointly selected, per intent, at inference time.

Third instance of the Composition Separation Principle. Part (1) is Theorem A's separation on the operating-point axis: a linear set of primitives spans a combinatorial frontier; static enumeration occupies sparse points. The principle now holds on three axes — intent coverage (Paper I), control-flow topology (§4), operating point (§6) — which is evidence the framework is discovered rather than assembled.

Honest scope. The operating points (a,ℓ,γ) must be estimated; like Proposition C's features and Theorem D's $\hat{q}_c$, they come from validation and carry estimation error. The frontier the system selects from is an *estimated* frontier — which is precisely what Proposition F accounts for.

6.1 Proposition F — Joint Configuration Selection

Proposition C selects a topology from the retrieved subgraph; Proposition E selects a model and budget from the operating constraint. The full harness configuration must ultimately be chosen *jointly*. We show the joint problem is tractable and regret-bounded, and we identify exactly when it decomposes back into Propositions C and E.

Configuration grid. Write $\mathcal{G}$ for the set of admissible configurations

$$h = (\text{topology}, \text{model}, k, \text{iteration budget}), \qquad |\mathcal{G}| = |\mathcal{H}| \cdot L \cdot K \cdot I.$$

Per intent i and operating constraint, each h has a true utility $U(h,i) \in [0,1]$ — accuracy when h is feasible and 0 otherwise, or a scalarization of the operating point. The system does not run every configuration; it scores them with an *operating-point predictor* $\widehat{U}(h,i)$ learned from validation, and selects $\widehat{h}(i) = \arg\max_{h \in \mathcal{G}} \widehat{U}(h,i)$. Let $h^\star(i) = \arg\max_{h \in \mathcal{G}} U(h,i)$ be the true optimum.

Proposition F.

(1) Constant grid. $|\mathcal{G}| = |\mathcal{H}| \cdot L \cdot K \cdot I$ is independent of the library size N_S and the number of intents M. Exhaustive per-intent selection over $\mathcal{G}$ is therefore $O(1)$ in problem size. (A fixed-agent system attaining the same configuration set would need $|\mathcal{G}|$ separately built agents — the Composition Separation Principle once more.)

(2) Regret bound. If the predictor satisfies $|\widehat{U}(h,i) - U(h,i)| \le \xi$ for all $h \in \mathcal{G}$, the selected configuration is 2ξ-optimal: $U(h^\star(i),i) - U(\widehat{h}(i),i) \le 2\xi$. If each cell's prediction averages n i.i.d. validation runs, then with probability $\ge 1-\alpha$, $\xi \le \sqrt{\frac{1}{2n} \log \frac{2|\mathcal{G}|}{\alpha}}$, so the selection regret is $O(\sqrt{\log |\mathcal{G}|/n})$ — vanishing in validation size and only logarithmic in the (already constant) grid.

(3) Decomposition. If topology preference is *model-invariant* — for every pair of topologies the sign of $U(t,m,\cdot) - U(t',m,\cdot)$ does not depend on the model m — then the joint optimum factorizes: choosing the topology by Proposition C and then the model and budget by Proposition E is exact. When this fails, the suboptimality of the decomposed (C-then-E) procedure is bounded by the maximum topology×model interaction $\iota = \max \big|[U(t,m) - U(t',m)] - [U(t,m') - U(t',m')]\big|$, and the constant-size joint pass of (1) removes it.

Proof. (1) $|\mathcal{G}|$ is a product of four knob cardinalities, none depending on N_S or M; the agent comparison is the now-familiar separation. (2) Writing $\widehat{h} = \widehat{h}(i)$ and $h^\star = h^\star(i)$,

$$U(h^\star) - U(\widehat{h}) \;\le\; [\widehat{U}(h^\star) + \xi] - [\widehat{U}(\widehat{h}) - \xi] \;=\; \widehat{U}(h^\star) - \widehat{U}(\widehat{h}) + 2\xi \;\le\; 2\xi,$$

since $\widehat{U}(\widehat{h}) \ge \widehat{U}(h^\star)$ by the choice of $\widehat{h}$. For the sample bound, Hoeffding gives $\Pr[|\widehat{U}(h) - U(h)| > \epsilon] \le 2e^{-2n\epsilon^2}$ per cell; a union bound over the $|\mathcal{G}|$ cells, set equal to α, yields $\epsilon = \sqrt{\log(2|\mathcal{G}|/\alpha)/(2n)}$. (3) Under model-invariance there is a topology $t^\star$ preferred under every model, so $\max_{t,m} U(t,m) = \max_m U(t^\star, m)$: selecting $t^\star$ by Proposition C and then optimizing the model and budget by Proposition E attains the joint maximum. When preference is not model-invariant, the decomposed choice differs from the joint optimum by at most one interaction term ι, by the definition of ι; exhaustive evaluation over the constant grid $\mathcal{G}$ incurs no such loss. ■

Remarks.

Why composition makes the joint problem easy. The joint configuration space is a *product* of small primitive sets, hence a small constant — typically a few hundred cells — so exhaustive scoring

per intent is trivial. The difficulty is not search but *scoring*: the operating-point predictor $\widehat{U}$. Proposition F localizes the entire practical difficulty into one estimable object and bounds the price of estimating it imperfectly. A fixed-agent system has no grid to search and no predictor to learn — only because it has pre-collapsed the choice, at the cost Proposition E(1) quantifies.

Decomposition is the common case, not the rule. Model-invariance of topology preference often holds approximately — a verification loop tends to help regardless of base model — in which case Propositions C and E compose exactly and the joint pass is unnecessary. The interaction term ι measures precisely how far a deployment sits from that regime, and is itself measurable (§9).

7 The Skill-Evolution Loop

Theorem D produces an abstention/warn log: a record of intents the system declined or flagged. That log is not waste — it is the system's demand signal for where competence must grow. AGC is the closure of an evolution loop that runs in two modes. *Extensive* evolution adds skills to serve demand the system cannot yet reach; *intensive* evolution optimizes the skills it already has against the demand they already serve. We give the extensive loop first.

1. **Collect.** Accumulate abstained and warned intents from the Theorem-D router.
2. **Cluster.** Group them into coherent unmet-demand regions; a sufficiently dense, coherent cluster is a candidate new taxonomy leaf.
3. **Localize.** For warn-region failures the answer was produced but unreliable, so the retrieved skills and the skill subgraph are known; the typed graph supports *credit assignment* — identifying which skill, or which composition edge, is implicated. This is the structural advantage over a monolithic agent, whose failures are unattributable.
4. **Mutate.** Propose a remedy: a new skill, a refinement of an existing skill, a split of an overloaded skill, or a new `requires`/`verifies` edge.
5. **Validate.** Test the candidate on a validation set for the affected leaf.
6. **Check impact.** Re-run retrieval-impact tests and regression tests on affected accepted leaves; a new skill or edge is not allowed to perturb old accepted regions without preserving their Theorem-D guarantee.
7. **Register or roll back.** If all gates pass, admit the change to the library and taxonomy; otherwise reject it or roll it back. Passing intents migrate from abstain/warn to answer only through this registry gate.

   Proposition (gated monotone frontier expansion). *If a loop iteration registers only after affected-leaf validation, retrieval-impact checks, and regression checks preserve every previously accepted leaf's Theorem-D guarantee, then the registered change weakly enlarges the answerable region $\mathcal{A}_\varepsilon$ or weakly improves the operating point of intents already inside it. Holding the validation distribution fixed, competence — the deployment mass inside $\mathcal{A}_\varepsilon$, weighted by reliability — is therefore monotone non-decreasing across accepted registry states, modulo distribution shift.*

The "modulo distribution shift" is essential and honest: as the world drifts, (D1) erodes and previously answerable intents can fall out of $\mathcal{A}_\varepsilon$. The loop therefore serves double duty — it expands the frontier *and* re-validates against drift. Competence is maintained, not banked. The monotonicity claim is about accepted registry states, not every proposed mutation.

Intensive evolution: skills as independently optimizable units. The same structure that localizes failures in step 3 also makes each skill *independently and automatically optimizable.* Because

every skill — every parent-skill subtree — is linked to a fixed set of intents in the taxonomy, and each of those intents carries a validation set, a skill has a well-defined, *stationary* fitness function: its validated success, latency, and cost over the intents it is responsible for. A stationary objective is the precondition for black-box optimization. The system can therefore run *evolutionary search* over a skill's parameterization — its instructions, its decomposition into sub-skills, its tool bindings, its compiled form — mutating and selecting variants by measured fitness, as a continuous background process rather than a reactive patch. Compilation (Paper I, §9) is one discrete move in this search: replacing an interpreted skill with validated code is a mutation scored against the same fixed intent set.

Two properties make this more than ordinary tuning. First, the objective is stationary *only because the intent linkage is fixed* — it is the decomposition into intent-bound skills that manufactures a stable fitness signal where the monolithic system has only a drifting, aggregate one. Second, improvements are admitted only if they pass regression gates: optimizing skill A against its intents cannot silently degrade skill B and remain registered, because B is guarded by its own validation set and its own acceptance test (Theorem D). A monolithic agent admits neither property — it is not decomposed into parts with separate objectives, so it cannot be evolved part-wise, and any change is an entangled change to the whole, regression-prone across every capability at once. Skill-to-harness assignment converts one large, non-stationary, non-decomposable optimization problem into many small, stationary, independently gated ones; that conversion is what makes automated self-improvement tractable, and it is a benefit no fixed-agent or monolithic system can claim.

Contrast with exploration-driven lifelong learning. Voyager's curriculum proposes novel goals to *explore* (Wang et al., 2023). This loop is *demand-driven*: its curriculum is the abstention log of real failed intents. Demand-driven growth aligns the system's improvement with economic value and makes "what to learn next" an observable, not a heuristic.

The boundary, stated plainly. This loop expands competence by *recombination and refinement within a human-seeded closure* — new compositions, refined skills, new edges. It does not invent genuinely new primitive capabilities from raw experience. That boundary is not a bug: it is exactly what keeps the taxonomy well-defined and abstention well-posed. It is also why §8 is careful about what AGC claims.

8 Artificial Generalizing Competence

We can now state the thesis precisely.

The prevailing target of the field is *Artificial General Intelligence*. Both load-bearing words are, for deployment purposes, problematic. *Intelligence* is not measurable: there is no task, no standard, no unit. *General* names a finished state — a system that can do anything — which is unfalsifiable and, as an engineering specification, vacuous.

We propose **Artificial Generalizing Competence** as the target a deployable system should actually be built and evaluated against, and the two word-swaps are the argument.

> **Definition (Artificial Generalizing Competence).** A system has *Artificial Generalizing Competence* over a domain when four conditions hold, each a measured quantity rather than an aspiration:
>
> 1. **Scoped competence.** There is an explicit, enumerated set of intents — the

competence frontier $\mathcal{A}_\varepsilon$ — on which the validated success rate is at least $1-\varepsilon$ at a stated statistical confidence (Theorem D).

2. **Declared boundary.** Intents outside $\mathcal{A}_\varepsilon$ are not failed silently; they are identified and routed to abstention or clarification. The system can state, and bound, what it does *not* reliably do (Theorem D corollary; §3, input boundary).
3. **Priced operation.** Every intent in $\mathcal{A}_\varepsilon$ carries a latency and a cost alongside its reliability, and the operating point is selectable within a declared budget (Propositions E, F).
4. **Monotone extension.** A validated loop weakly enlarges $\mathcal{A}_\varepsilon$ and improves the operating points within it over time: competence has a positive, measured *rate of generalization*, and never a claim of completion (§7).

Equivalently: a system has AGC exactly when Theorem D, Proposition D-P, Propositions E and F, and the §7 loop hold of it. AGC is *defined to be constructible and checkable* — that is the whole of the contrast with AGI.

This is **earned autonomy**: the system is allowed to act autonomously only over regions where competence has been validated, priced, and permissioned. Outside that earned region, it clarifies, warns, abstains, or refuses authorization. Autonomy is therefore not a personality trait of the model; it is an operational status granted by the frontier $\mathcal{A}_\varepsilon$ and revoked when validation or permissions no longer support it.

The contrast with Artificial General Intelligence is then exact, dimension by dimension:

	AGI, as usually defined	AGC, as defined above
Promise	eventually, any task a human can do	success on a named, enumerated intent set $\mathcal{A}_\varepsilon$
Measurement	no task, no standard, no unit	a validated success rate with a confidence interval
Cost	unspecified	a latency and a price per intent, budget-selectable
Boundary	none — a system that does "anything" cannot say what it cannot	explicit — the complement of $\mathcal{A}_\varepsilon$, abstained on or clarified
Status	terminal — a finish line, possessed or not	processual — a frontier with a measured rate of extension
To a buyer	a claim that cannot be audited, priced, or contracted	every clause can be audited, priced, contracted, monitored

Each AGI row is a predicate no business can verify before purchase or enforce after it. Each AGC row is a number that can go into a contract. That is the benefit — and it is not that AGC is more modest than AGI, but that AGC is *specified*. An enterprise can procure AGC, govern it, and hold a vendor to it; "AGI" names nothing an enterprise can sign.

Competence, not intelligence. Competence is always competence *at* something, *to* a standard. It is therefore measurable — and the validation sets skill-to-harness assignment already requires are the standard. Theorem D turns competence into a number with a confidence interval; Proposition E turns it into a number with a price. Intelligence admits neither operation.

Generalizing, not general. "General" is an adjective asserting a completed property. "Generalizing" is

a present participle naming a process — the §7 loop, which extends the answerable frontier through validated registry updates and improves the competence already inside it. An AGC system never claims to be done. Its calibrated "I cannot reliably answer that yet" is not an apology; it is the honest, machine-checkable report of where the frontier currently lies.

This is the sharp distinction. An AGI claim asserts *generality*; it cannot, even in principle, exhibit a credible boundary. An AGC system exhibits its boundary *by construction* — that boundary is $\mathcal{A}_\varepsilon$ — and commits to moving it. The presence of a well-defined "cannot" is the proof that the system is AGC and not making an AGI claim.

AGC is what deployment needs — not AGI plus guardrails. The dominant commercial narrative proposes to build a maximally general intelligence and then *operationalize* it: bolt on guardrails, reliability layers, refusal policies, cost controls, and access rules after the fact. The results of this paper say that ordering is backwards. Governance retrofitted onto an undecomposed general model is governance without a foundation. Theorem D's corollary is exactly the statement that abstention added to a system with no explicit, validated partition of its task space *cannot* carry a coverage guarantee — the guardrail may fire, but nothing bounds what it lets through. The same gap applies to cost, which Proposition E governs only because the configuration space is composed and exposed, and to permissions, which Proposition D-P can gate only because the system is built from skills. In an AGC system these are not retrofits; they are load-bearing structure — the validated taxonomy, the operating-point frontier, the permission graph — present before the first deployment. A business does not need a general intelligence that must afterwards be made safe and operable; it needs a system that is *governable by construction* and *measurably competent* on the intents it claims, and calibrated about the rest. In the model analyzed here, the guardrail-retrofit pattern reaches the same guarantees only by adding the same validated partition, permission graph, and operating frontier that define AGC. This does not deny weaker calibration or abstention guarantees for systems without an explicit taxonomy; the claim is that they do not obtain the Theorem D-strength, leaf-conditioned coverage–risk guarantee without an equivalent validated partition.

We therefore do not argue that Athanor is a path to AGI, and we think most "we need AGI to automate this" claims are a category error: what such tasks actually require is broad, validated, composable skill coverage with calibrated abstention on the complement — an *engineering* target, the one this paper and Paper I formalize. AGC is that target named honestly. Whether sufficiently broad AGC eventually becomes indistinguishable from what people mean by AGI is a question we deliberately leave outside the formal claims; nothing in this paper depends on the answer.

9 Evaluation Protocol

Proposition C — harness selection. Construct intents whose retrieved subgraphs span distinct structural regimes (purely parallel, deep-sequential, verification-heavy, single-skill). Compare each fixed topology against the structural-readout rule. *Predicted:* no fixed topology is best across regimes; the readout rule weakly dominates each and strictly dominates under a structurally mixed intent distribution when (C1) holds. Ablate the feature map to identify which structural features carry the decision.

Theorem D — selective reliability. Build a taxonomy with per-leaf validation sets. Sweep the risk tolerance ε and plot the empirical coverage–risk curve: conditional failure rate on $\mathcal{A}_\varepsilon$ against ε, and abstention rate against ε. *Predicted:* conditional failure tracks at or below the diagonal ε.

Stress (D1) by deliberately coarsening the taxonomy and show the guarantee degrades, quantifying the granularity tradeoff. Inject out-of-taxonomy intents and confirm the $\kappa \geq \tau$ gate abstains rather than mis-accepting. Separately, present under-specified inputs and verify that boundary clarification raises κ and recovers intents that would otherwise abstain.

Proposition D-P — authorization soundness. Construct matched authorized and unauthorized intents with identical task semantics but different user/context permissions. Confirm that authorized intents can enter the answer region only when their validation bounds pass, while unauthorized intents are routed to abstain/refuse even when the underlying model would otherwise comply. *Predicted:* no emitted answer invokes an unauthorized skill, tool, or datum; failures indicate a policy-graph or executor-enforcement bug, not a statistical calibration error.

Proposition E — operating-point frontier. Trace the achievable (a, ℓ, γ) region by sweeping configurations; overlay the sparse scatter realizable by a comparable fixed-agent suite. *Predicted:* the composition frontier dominates and is far denser. Then fix intents and vary the budget B to exhibit budget-dependent abstention — intents crossing from warn to answer as B rises — and trace per-intent reliability–cost curves.

Proposition F — joint configuration selection. Hold out validation runs to fit an operating-point predictor $\widehat{U}$; measure its sup-error ξ over the configuration grid $\mathcal{G}$; confirm the selected configuration's realized utility is within 2ξ of the grid-exhaustive optimum. Estimate the interaction term ι to determine whether C-then-E decomposition is exact for the deployment or whether the joint pass is needed. *Predicted:* regret tracks the $O(\sqrt{\log|\mathcal{G}|/n})$ bound and shrinks with validation size.

Skill-evolution loop. Seed an abstention log, run the *extensive* loop, and measure (i) the fraction of abstained intents that migrate to answer after validated registration, and (ii) that re-validation holds the Theorem-D guarantee on the enlarged $\mathcal{A}_\varepsilon$. Separately, run *intensive* evolutionary search on a chosen skill against its fixed intent-validation set, and measure (iii) validated improvement in that skill's operating point and (iv) the absence of regression on other skills' validation sets. *Predicted:* monotone frontier expansion and per-skill improvement, with the guarantee preserved and improvements composing without regression.

10 Discussion and Limitations

Estimation error is pervasive. Proposition C, Theorem D, and Proposition E all consume validation-derived estimates — structural error rates, per-leaf success rates, operating points. The guarantees are conditional on those estimates; in particular Theorem D's $1 - \alpha$ confidence is exactly the honest accounting of validation sample size. None of the results claim more than the validation data supports.

The taxonomy and ontology are assumed, not derived. This paper takes the domain ontology, the intent taxonomy, and the skill graph as given. Constructing and maintaining them — curating canonical definitions, choosing leaf granularity against the (D1) tradeoff — is itself substantial work. §7's loop grows the taxonomy at its frontier but does not bootstrap it from nothing.

Harness selection composes. Proposition C selects topology, Proposition E selects model and budget, and Proposition F shows the *joint* selection over the full configuration is a constant-size, regret-bounded optimization — exact when topology preference is model-invariant, and otherwise resolved by a cheap joint pass over the grid. What remains is practical rather than formal: learning

the operating-point predictor $\widehat{U}$ that Proposition F consumes, and measuring its error ξ and the interaction term ι on a real deployment (§9).

AGC is bounded. As §7 and §8 state, the evolution loop is recombination within a seeded closure. A system that must serve genuinely novel primitive capabilities — not new compositions of existing ones — falls outside the model, and AGC correctly abstains on it. We regard this as a feature, but it is a real limit on the scope of the claims.

Relation to Paper I. Paper I's Theorems A and B are prerequisites: they justify the skill-to-harness assignment architecture this paper operates. Proposition C, Proposition E(1), and Proposition F(1) are, with Paper I's Theorem A, four instances of the Composition Separation Principle as a design pattern. Theorem D, Proposition D-P, and the §7 loop are specific to the governance layer and have no analogue in Paper I.

11 Conclusion

Paper I established the conditions under which skill-to-harness assignment is the architecture to build. This paper showed how to *run* it as a governable system. Proposition C makes control-flow topology a structural readout of the retrieved skill subgraph under a stated loss model. Theorem D makes calibrated abstention a coverage–risk guarantee, available only to architectures with an explicit validated intent taxonomy, and silently impossible without one. Proposition D-P adds the narrower authorization guarantee: the system cannot answer through resources the user may not access. Proposition E makes the accuracy/latency/cost operating point a composed, user-selected object, and reveals that abstention is budget-dependent — so the system can hand a user a per-intent reliability–cost curve rather than a fixed verdict. The abstention log then closes a demand-driven loop that extends the answerable frontier through validated registry updates.

We named the target **Artificial Generalizing Competence**: competence because it is measurable against a standard, generalizing because it provably extends, and neither general nor intelligence because the system always exhibits — and commits to moving — an explicit, honest boundary. The calibrated "I cannot reliably answer that yet" is not the system falling short of the goal. It is the goal, correctly stated.

A Empirical Evidence Package

> **Empirical status.** The live CRB evidence uses real `gpt-4.1` and `text-embedding-3-large` calls. The current Paper II support is targeted rather than full-suite: CRB-055 supports the D/E/F/G operating claims over `5,360` live events, CRB-072 and CRB-086 support ontology benefit under realistic overlays and hard distractors, and CRB-037 supports tool-selection precision. The answer-success overlay-collapse direction remains refuted: flat-MCP can still answer correctly while over-selecting. When this paper relies on Paper I's retriever premise, it uses the repaired CRB-089/090 `hybrid_faceted` path, which reaches `0.992` coverage at `N_S=1600` with `lambda_hat=0.337`. Proposition D-P is authorization soundness from a specified permission graph; it is not a policy correctness or executor-audit theorem.

The current CRB evidence package is maintained in `reports/crb052_paper_evidence_package.md`. The table below records the empirical support this draft should cite.

Claim	Current evidence	Paper use
Proposition C harness selection	CRB-006 medium live; CRB-045 corrected aggregation	Supported when split structural regimes are aggregated.
Theorem D selective reliability	CRB-055; CRB-042; CRB-045	Selective reliability, not high acceptance coverage at every tolerance.
Proposition D-P authorization soundness	CRB-055; CRB-045 forced unauthorized condition coverage	Authorization safety support only; not an accuracy or policy-correctness claim.
Proposition E operating frontier	CRB-055; CRB-041; CRB-045	Integrated targeted support; full all-paper suite deferred.
Proposition F joint configuration regret	CRB-055; CRB-041; CRB-045	Supported with measured regret `0.055`; predictor `xi=0.099`.
Skill-evolution loop	CRB-055; CRB-041; CRB-045	Synthetic loop support, not production lifecycle evidence.
Ontology benefit	CRB-024; CRB-045 skill-to-harness assignment rows; CRB-072 and CRB-086 realistic ontology overlays	Controlled synthetic alias-resolution support plus realistic hard-distractor overlay confirmation.
Overlay precision	CRB-037; CRB-045 overlay	External-validity precision support only.
Overlay answer-success degradation	CRB-045; CRB-029/037 diagnostics	Unsupported limitation; flat-MCP can still answer correctly while over-selecting.

A.1 Integrated D/E/F/G Exhibit

Claim	CRB-055 result	Figure asset
Theorem D selective reliability	`0` coverage-risk violations; condition coverage complete across an `M=4` taxonomy support and eight risk/validation cells with `validation_per_leaf` in $\{10, 50\}$ and `100` live test events per cell.	`figures/live_crb055/live_crb055_e_d_risk_repaired_integrated/theorem_d_coverage_risk.svg`
Proposition E operating frontier	`8` Pareto points across `72` configurations.	`figures/live_crb055/live_crb055_e_e_frontier_integrated/proposition_e_pareto_frontier.svg`
Proposition F joint configuration regret	Regret `0.055`; predictor sup-error `xi=0.099`; bound satisfied.	`figures/live_crb055/live_crb055_e_f_joint_config_integrated/proposition_f_regret.svg`
Skill-evolution loop	Migration, monotonicity, and regression-free flags all `1.000`.	`figures/live_crb055/live_crb055_e_g_evolution_integrated/paper_ii_evolution_loop.svg`

CRB-055 is the targeted integrated Paper II confirmation. It is not a full all-paper suite.

A.2 Ontology Benefit Exhibit

Evidence layer	Result	Paper use
Controlled synthetic mechanism	CRB-024 ontology benefit `not_refuted`; absent-minus-injected effect `0.950`; perturbed-minus-injected effect `0.958`; `chi_sem=0.958` with 95% CI `[0.933, 0.984]`.	Main mechanism support for shared ontology reducing semantic mismatch.
Realistic overlay confirmation	CRB-072 ran `396` events, full per-substrate condition coverage, and `52/52` applicable directions passed. CRB-086 added `1,188` hard-distractor events at counts `0`, `48`, and `96`, with ontology-specific checks `120/120` passed.	Bounded external-validity and hard-distractor support.
Strongest realistic effect	In CRB-086, injected-vs-absent answer margins are `+0.806`, `+0.833`, and `+0.833` at `0`, `48`, and `96` hard distractors; injected-vs-perturbed answer margins remain `+1.000` at every count.	Shows injected ontology prevents semantic drift from absent or divergent local definitions even with plausible neighboring tools.
Scope limitation	CRB-086 is still a fixture-based overlay and does not make the full all-paper suite complete.	Do not present as broad public-suite coverage or universal flat-tool answer collapse.

A.3 Realistic Overlay Precision Exhibit

Evidence	Result	Paper use
CRB-037 confirmation	`1200` events, `9/9` precision checks passed, no warnings.	Main realistic-overlay precision support.
Final hard-distractor stratum	At `96` hard distractors, flat-MCP exact selection `0.967` with over-selection `0.033`; skill-to-harness exact selection `0.983` with over-selection `0.017`.	Supports exact-selection/over-selection precision, not answer-collapse.
Unsupported direction	Realistic overlay answer-success direction remains `refuted`.	State that flat-MCP can still answer correctly while selecting plausible extra tools.

When this paper cites Paper I's robustness mechanism, it should cite CRB-017/021 for Theorem B's progressive-disclosure mechanism, CRB-089/090 for the faceted high-library retriever repair, and CRB-028/045 for the E-B4a no-echo advantage. It should not use E-B4b descriptor echo as main support; CRB-049 makes that an appendix/control artifact.

Current paper-bundle decision artifacts are CRB-056 through CRB-155: CRB-056 aligns current documents to CRB-055, CRB-057 scopes architecture claims to the formal assumptions, CRB-058 keeps the full suite cost-deferred, CRB-120/155 ran the Theorem B reviewer-risk fallback with a mixed same-suite result that does not upgrade Paper I, CRB-060 verifies synthetic-provenance labeling, and CRB-071/072 repair and confirm the realistic ontology-overlay path. CRB-073 defers further paid work absent a reviewer-triggered gap, CRB-074 through CRB-077 assemble and insert the core exhibits, CRB-086 adds hard-distractor ontology support, CRB-089/090 close Paper I's faceted retriever gate, CRB-114/115 align the evidence posture, and CRB-116 through CRB-155 cover manuscript readiness, private-review/counsel handoff, permission-graph formalization, claim traceability, packet-currency checks, current PDF build verification, readiness gating, counsel-intake completeness auditing, and the mixed same-suite Theorem B reviewer-risk probe before patent-counsel intake or public release review.

References

Bibliographic details and affiliation metadata to be finalized.

- Tan, G. (2026). GBrain: Garry's Opinionated OpenClaw/Hermes Agent Brain. GitHub repository. https://github.com/garrytan/gbrain.
- McCord, A. (companion). Composition Beats Bundling: A Separation Theorem for Late-Bound LLM Agents. — *Paper I.*
- Angelopoulos, A. N., & Bates, S. (2023). Conformal Prediction: A Gentle Introduction. *Foundations and Trends in Machine Learning.*
- Anthropic (2025). Equipping Agents for the Real World with Agent Skills. *Anthropic Engineering Blog.*
- Chen, L., Zaharia, M., & Zou, J. (2023). FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv:2305.05176.*
- Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., & Mordatch, I. (2023). Improving Factuality and Reasoning in Language Models through Multiagent Debate. *arXiv:2305.14325.*
- Geifman, Y., & El-Yaniv, R. (2017). Selective Classification for Deep Neural Networks. *NeurIPS.*
- Hu, Q. J., Bieker, J., Li, X., Jiang, N., Keigwin, B., Ranganath, G., Keutzer, K., & Upadhyay, S. K. (2024). RouterBench: A Benchmark for Multi-LLM Routing System. *arXiv:2403.12031.*
- Kirichenko, P., Ibrahim, M., Chaudhuri, K., & Bell, S. J. (2025). AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions. *arXiv:2506.09038.*
- Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., & Stoica, I. (2024). RouteLLM: Learning to Route LLMs with Preference Data. *arXiv:2406.18665.*
- Shinn, N., *et al.* (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. *NeurIPS.*
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv:1712.01815.*
- Snell, C., Lee, J., Xu, K., & Kumar, A. (2024). Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters. *arXiv:2408.03314.*
- Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic Learning in a Random World.* Springer. (Conformal and Mondrian conformal predictors.)
- Wang, G., *et al.* (2023). Voyager: An Open-Ended Embodied Agent with Large Language

Models. *arXiv:2305.16291*.
- Wen, B., Yao, J., Feng, S., Xu, C., Tsvetkov, Y., Howe, B., & Wang, L. L. (2024). Know Your Limits: A Survey of Abstention in Large Language Models. *TACL*. arXiv:2407.18418.
- Yao, S., *et al.* (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *ICLR*.
- Anonymous. (2025). Calibrating LLMs for Selective Prediction: Balancing Coverage and Risk. *OpenReview / NeurIPS submission*.

End of draft. Open items: (1) finalize affiliation and venue metadata; (2) strengthen the operating-point predictor $\widehat{U}$ of Proposition F and reduce measured ξ beyond the current CRB-055 integrated run; (3) extend the targeted CRB-055 D/E/F/G evidence into a full all-paper protocol only if reviewer or venue demand justifies the cost. The coverage–risk curve for Theorem D should still anchor the empirical section, with the CRB-055 selective-reliability caveat. The shared model and notation are aligned with Paper I (§3).