TOWARDS A THEORETICAL FOUNDATION FOR TOOL-AUGMENTED LLM SKILL LIBRARIES: THE PRICE OF COMPOSITION, PROPAGATION, AND SAFETY UNDER EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill libraries, natural-language procedures that condition a frozen large language model (LLM), carry tools, and read and write persistent artifacts, are now a de facto standard for building agents. Their reliability, however, still rests on folklore: we cannot yet say how much return an agent loses by composing skills, or how long a safety guarantee survives as the library is edited. We construct a single end-to-end object that answers both: a *certificate* for a composed agent that bounds the return it loses to composition, measured against the best its frozen model could achieve on a fixed plan, with every term one the designer can drive down or measure and which stays valid over an explicit editing horizon. Two modeling choices make the bound well-posed. First, gaps are measured against the *realizable* optimum, the best conditioning of the fixed model on the given plan, quarantining base-model capability and plan structure as invariants. Second, a two-layer model separates a within-invocation decision process from a cross-invocation calculus on artifacts. We then assemble the certificate term by term: an exact decomposition names where return is lost; a *propagation theorem* bounds how per-skill errors compound along the plan, letting each skill *rupture* (fail discontinuously) with bounded probability rather than assuming Lipschitz behavior, and yielding a convergent/linear/exponential depth trichotomy with matching lower bounds, a checkpoint rule, and black-box-measurable constants; size and learnability bounds control the errors at their source; a two-regime safety result supplies the horizon; and a capability-flow margin demotes tool-combination danger to higher order in the rupture rate. Five algorithms make the certificate constructive. Empirically, a black-box harness recovers the propagation constants on a 332-skill, six-database SKILL.md suite; across five models and six domains the resulting deploy/reject verdict discriminates on both axes; and on real end-to-end LLM pipelines over a nine-model sweep the measured loss stays within the certificate in every case.

1 INTRODUCTION

A *skill* is a natural-language document with a trigger and a body; matching skills are retrieved by embedding similarity and concatenated into a frozen model's context, steering behavior with no weight update. In the last two years this pattern has become a standard substrate for LLM agents: open formats such as SKILL.md Anthropic (2025) and the Model Context Protocol Anthropic (2024) package reusable procedures and tools; retrieval pipelines select among tens of thousands of community skills Li et al. (2025); and benchmarks report that curated skills lift agent success rates by double digits Li et al. (2026). Real skills carry tools Schick et al. (2023); Yao et al. (2023b), operate on persistent artifacts, and are composed: several run per query, some in parallel and some consuming others' outputs. Yet this machinery has almost no theory. We lack answers to how a library loses return, how composing skills amplifies error, how many skills a domain needs, and how safety drifts as a library is edited.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The destination: an agent certificate. For an agent that executes a *plan* of retrieved skills (a DAG of invocations) against a library $\mathcal{L}$, we want a guarantee

$$J_\tau^\circ - \mathbb{E}[J_\tau(\text{agent})] \;\le\; \underbrace{B(t)}_{\text{composition}} \;+\; \underbrace{P^{\text{rup}}(t)}_{\text{ruptures}}, \qquad t \le T^\star, \tag{1}$$

where the left side is the return the agent loses against the best its frozen model can do. The right side is built entirely from quantities the designer controls: retrieval coverage, per-skill authoring quality, per-skill sensitivity and *rupture* rate, and checkpoint placement, each designed down or measured from the black-box model, with the bound holding up to a horizon $T^\star$ over which it survives library edits. Here $B(t)$ is a composition bound and $P^{\text{rup}}(t)$ a rupture budget (Section 5, Theorem 1); the paper constructs the right-hand side of (1) and the synthesis (Section 8, Proposition 2) assembles it. Two modeling problems must be settled first, or the left side is not well-defined.

Problem 1: against what optimum? Gaps are meaningless against the unconstrained MDP optimum J^*, which folds in capability the fixed model lacks. We instead measure every gap against the *realizable* optimum J°, the supremum over conditionings of the fixed model, grounded in the in-context-learning view of conditioning as policy selection Xie et al. (2022); Bai et al. (2023); Li et al. (2023); the remainder $J^* - J^\circ$ is quarantined as a library-invariant *capability gap*, so every gap we decompose is genuinely improvable by editing the library.

Problem 2: what is the right abstraction? A single decision process does not fit a pipeline of separate invocations exchanging artifacts. We use a *two-layer* model: *within* an invocation, a discounted decision process over tokens and tool calls; *across* invocations, a behavioral quality calculus on artifacts. The discount γ drives scheduling, while cross-skill error propagation is a γ-free statement about artifact quality.

Contributions.

- A well-posed *realizable-gap* target and an *exact decomposition* of it into retrieval, selection, interference, and tool terms, with a reduction lemma putting every per-skill error in one common unit (Sections 3–4).
- A *propagation theorem* for non-Lipschitz skills (Theorem 1), pairing a stable-manifold bound with a rupture budget, with a depth trichotomy (Corollary 1), matching lower bounds (Theorem 5), a checkpoint rule (Corollary 2), and constants recovered black-box by perturbation. This is the technical core.
- Library *size and learnability* bounds (Theorem 2) and a two-regime *safety* result giving the horizon $T^\star$ (Theorem 3), with a *compositional* analysis: single-skill danger cannot bound joint danger (Proposition 1), and a capability-flow margin demotes joint danger to order $\lceil k/c \rceil$ in the rupture rate (Theorem 4).
- A *synthesis* into one inequality (Proposition 2), five algorithms making it constructive (Section 9), and validation on a real 332-skill, six-database suite plus a live grid of five models $\times$ six domains on which the verdict discriminates by both axes (Section 10, Appendix G).

Each result names what it *consumes* and *produces* (Fig. 1); the chain is the contribution. Notation is in Appendix A (Table 2); proofs and secondary results in Appendices B–C.

2 Related Work

The machinery of tool-augmented skill libraries has been assembled quickly and largely empirically. Reusable-skill agents grew from executable skill stores Wang et al. (2024) and reasoning-and-acting scaffolds Yao et al. (2023b); Shinn et al. (2023); Yao et al. (2023a), and standardized around open formats and protocols Anthropic (2025; 2024), retrieval over large skill and tool collections Li et al. (2025); Shi et al. (2025), declarative and tool-use pipelines Khattab et al. (2024); Schick et al. (2023), and benchmarks reporting success-rate lifts Li et al. (2026); the compound AI systems framing Zaharia et al. (2024) names the shift. This work shows skills help and standardizes how to ship them, but it is architectural, empirical, or definitional: for a deployed agent that composes skills it cannot say how much return composition costs, how many skills a domain needs, or how long a safety guarantee survives editing. Those are the questions we answer.

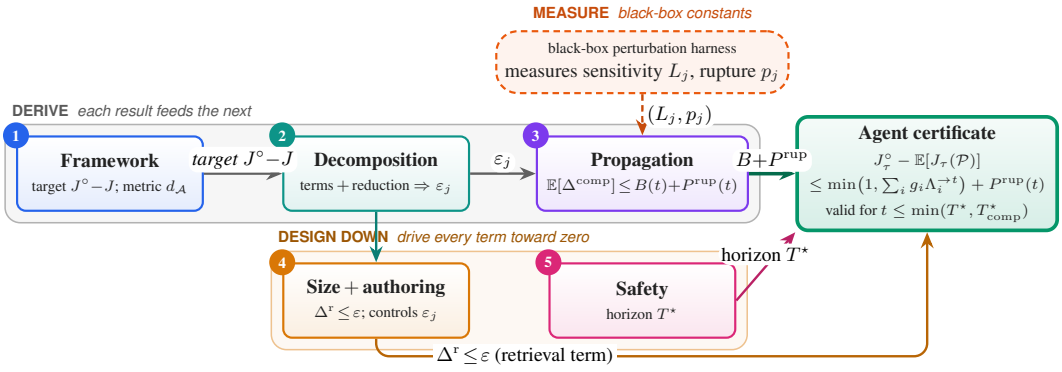


Figure 1: The argument as a dependency chain. The framework fixes the target gap; the decomposition expresses each skill's error ε_j in that target's units; propagation turns the ε_j into a bounded composition term $B(t) + P^{\mathrm{rup}}(t)$ whose constants are measured by a black-box perturbation harness; size and authoring bound the retrieval term and the inputs to ε_j; safety supplies the horizon $T^\star$; synthesis assembles all of these into the certificate (1).

Compounding error. That per-step error compounds is old and sharp in homogeneous settings: imitation-learning reductions Ross & Bagnell (2010); Ross et al. (2011) and model-based RL rollout bounds Janner et al. (2019); Asadi et al. (2018) assume a single repeated dynamics with a uniform Lipschitz constant and no discontinuities, which LLM skills violate node by node. On the language side the effect is documented but not modeled as a controllable quantity: Dziri et al. (2023) prove exponential error growth on computation graphs, Gan et al. (2025) formalize the reasoning-chain snowball, and others measure uncertainty propagation and multi-agent failure Duan et al. (2025); Cemri et al. (2025); Chen et al. (2024). The compositionality gap Press et al. (2023); Lake & Baroni (2018); Hosseini et al. (2022); Sakai et al. (2025) is a static failure ratio at fixed hop count, not a depth-indexed growth law. Our checkpoint rule turns on whether verifiers catch errors, and that evidence cuts both ways: verifiers and self-refinement help Cobbe et al. (2021); Lightman et al. (2024); Uesato et al. (2022); Madaan et al. (2023), yet self-correction is fragile without an external signal Huang et al. (2024a); Kamoi et al. (2024).

Optima, learnability, and safety. We read conditioning as selecting a policy within a realizable class, a view supported by the statistical theory of in-context learning Xie et al. (2022); Garg et al. (2022); Bai et al. (2023); Li et al. (2023), and measure gaps in the agnostic best-in-class sense Kearns et al. (1994) using behavioral pseudometrics from the bisimulation lineage Ferns et al. (2011); Castro et al. (2021). On safety, empirical work shows agents grow less safe as they evolve and acquire tools Shao et al. (2025); Gao et al. (2026); Vijayvargiya et al. (2025); Debenedetti et al. (2024) but stops at demonstration, offering no horizon. Defensive methods are either single-turn Sharma et al. (2025); Greshake et al. (2023) or runtime, intercepting capability and information flow during execution Debenedetti et al. (2025); Costa et al. (2025); the classical static analyses they build on Denning (1976); Dennis & Van Horn (1966) are deterministic and carry no rupture rate.

This paper. We turn this qualitative picture into a quantitative guarantee for a composed agent. Our propagation theorem departs from the precedents above on four axes at once: a heterogeneous plan DAG with per-edge sensitivities; a convergent/linear/exponential trichotomy with matching lower bounds; a rupture term for discontinuous failures in place of a Lipschitz assumption; and constants a practitioner can recover black-box. With the size, safety, and capability-flow results, one certificate bounds both the return an agent loses to composition and the danger it accrues, over an explicit editing horizon, every term designed down or measured. An extended survey is in Appendix F.

3 Framework

This section fixes the two objects (1) is stated in: the *target gap* $J^\circ - J$ and the *artifact pseudometric* $d_{\mathcal{A}}$ in which all later error is measured. (Every symbol is collected in Table 2, Appendix A.)

Agent, tasks, skills. Fix a vocabulary V and a frozen LLM $\Omega\colon V^* \to \mathcal{P}(A)$ with $A = A_{\text{txt}} \sqcup A_{\text{tool}}$. The state is $s_t = (c_t, W_t)$: context $c_t \in V^*$ and workspace $W_t \in \mathcal{W}$ (persistent artifacts). A *tool* is a stochastic kernel $\phi\colon (c_t, W_t, \text{args}) \mapsto$ a distribution over (result y, W_{t+1}); issuing it appends y to context and updates the workspace. A *skill* $\sigma = (N_\sigma, T_\sigma, B_\sigma, \Phi_\sigma, \Delta_\sigma)$ has name, trigger, body, tools, and dependencies; its conditioned policy is $\pi_\sigma(\cdot \mid s_t) = \Omega(\cdot \mid \text{sys}(\sigma) \oplus c_t)$ with $\text{sys}(\sigma) = N_\sigma \oplus T_\sigma \oplus B_\sigma$. A *task* $\tau = (r_\tau, \mu_\tau)$ with the shared dynamics and discount $\gamma \in [0, 1)$ defines a discounted MDP with return $J_\tau(\pi) = \mathbb{E}_{\mu_\tau, \pi}[\sum_t \gamma^t r_\tau]$, normalized to $[0, 1]$.

Realizable optimum and the quarantined capability gap. Fix a description-length budget ℓ (a token bound on a skill body, enforced by finite context). Let Σ_ℓ be the well-formed skills with $|B_\sigma| \le \ell$, $\Pi_\Omega = \{\pi_\sigma : \sigma \in \Sigma_\ell\}$ the *realizable policy class* ($|\Pi_\Omega| \le |V|^{\ell+1}$), and $J^\circ_\tau = \sup_{\pi \in \Pi_\Omega} J_\tau(\pi)$ the *realizable optimum*. The *capability gap* $\Delta^{\text{cap}}(\tau) := J^*_\tau - J^\circ_\tau \ge 0$ is library-invariant and set aside. Bounding ℓ keeps Δ^{cap} from collapsing (unbounded prompts give $J^\circ \to J^*$); Proposition 3 (Appendix B) exhibits a task family with $\Delta^{\text{cap}} \ge \frac{1}{2}$ for every finite ℓ, so the split is not vacuous. We take J° *plan-relative*, with plan-structure suboptimality Δ^{plan} quarantined alongside Δ^{cap}.

Two-layer model and the artifact pseudometric. Within an invocation, $((c_t, W_t), A, \gamma)$ is a discounted decision process terminating by emitting an output artifact. Across invocations, a plan composes artifacts, compared per plan slot, which is what makes the comparison a genuine pseudometric:

$$d^{(j)}_{\mathcal{A}}(a, a') := \sup_{\kappa \in \mathcal{K}_j} \left| J^\kappa(a) - J^\kappa(a') \right|, \tag{2}$$

where $\mathcal{K}_j$ is the plan's fixed downstream sub-DAG of slot j (with the task reward) and $J^\kappa(a)$ is the realizable return of running continuation κ when slot j holds a. Since $\mathcal{K}_j$ depends on the plan and slot (not on which candidate fills it), two candidates for node j share one continuation set, and $d^{(j)}_{\mathcal{A}}$ is a pseudometric (Lemma 3, Appendix B) measuring an artifact's worst-case downstream return impact in the same $[0, 1]$ units as the gaps. The task pseudometric is $d_{\mathcal{T}}(\tau, \tau') := \sup_{\pi \in \Pi_\Omega} |J_\tau(\pi) - J_{\tau'}(\pi)|$. A query Q is served by a plan $\mathcal{P} = (\{q_i\}, D)$ with data edges $q_i \to q_j$; $\text{pa}(j)$ are j's data-parents (Algorithm 1, Appendix C). For any policy x, its *realizable gap* is $\Delta(x, \tau) := J^\circ_\tau - J_\tau(x) \ge 0$; $\text{Ret}(\tau) \subseteq \Sigma_\ell$ is the retrieved set and σ_{sel} the selected skill.

Assumptions. We normalize returns to $[0, 1]$ (**A1**); assume bounded task complexity $M(\varepsilon) := \mathcal{N}(\varepsilon, \mathcal{T}, d_{\mathcal{T}}) < \infty$ (**A2**); and *realizability*: some body achieves gap $\le \varepsilon_0$ on every supported task (**A3**). The key modeling assumption replaces global Lipschitzness, which LLMs violate, by local sensitivity on a stable event plus bounded-probability ruptures.

Assumption 1 (Local sensitivity with ruptures)**.** Each node j has *per-edge* sensitivities $L_{i \to j} \ge 0$ (one per data-parent), intrinsic error $\varepsilon_j \ge 0$, and *rupture probability* $p_j \in [0, 1]$. There is a *stable event* S_j with $\mathbb{P}[S_j] \ge 1 - p_j$ on which, in the slot metrics (2),

$$\delta_{\text{out}}(j) \;\le\; \varepsilon_j + \sum_{i \in \text{pa}(j)} L_{i \to j}\, \delta_{\text{out}}(i),$$

$$\delta_{\text{out}}(j) := d^{(j)}_{\mathcal{A}}(a^{\text{real}}_j, a^{\text{ideal}}_j),$$

the deviation of node j's realized output from its ideal. On the *rupture* event S^c_j only $\delta_{\text{out}}(j) \le 1$ holds. Reward depends on the sink with $L_r = 1$. The uniform case writes $L_{i \to j} \equiv L_j$; setting $p_j \equiv 0$ recovers the globally-Lipschitz model. A skill invoking a tool ϕ has $p_j = 1 - (1 - p^{\text{LLM}}_j)(1 - p^{\text{tool}}_j)$, two *separately measurable* channels (perturb prompt vs. tool input).

Ruptures (format collapse, refusal, hallucination cascade) are the artifact-quality analogue of the safety *jumps* of Section 7, and are governed by the same continuous-vs-jump structure.

4 Exact Decomposition of the Realizable Gap

We name where realizable return is lost by a selected skill, so the rest of the paper can bound each term. Running a skill at three idealization levels (ideal tools/helpers ($\sigma^{IT,IH}$), real tools but ideal helpers ($\sigma^{H,IH}$), and real both (σ), each an *in-class maximizer* so every level stays $\le J^\circ_\tau$) gives an idealization ladder whose six in-class returns telescope into one exact identity.

Lemma 1 (Exact decomposition). *$J^*_\tau - J_\tau(\sigma_{\mathrm{sel}}) = \Delta^{\mathrm{cap}} + \Delta^{\mathrm{int}} + \Delta^{\mathrm{r}} + \Delta^{\mathrm{e}} + \Delta^{\mathrm{tool}} + \Delta^{\mathrm{help}}$, every term nonnegative; the last five sum to the controllable gap $J^\circ_\tau - J_\tau(\sigma_{\mathrm{sel}})$. Here Δ^{int} (interference: cost of a single fixed body across contexts), Δ^{r} (retrieval), Δ^{e} (selection), Δ^{tool} (tool), and Δ^{help} (helper input) are the successive ladder rungs.*

The interference rung is bounded by a worst-context regret, $\Delta^{\mathrm{int}} \le \nu := \min_{b_0} \sup_z [\max_b J(b; z) - J(b_0; z)]$ (Proposition 4, Appendix B). The bridge to propagation is the *reduction lemma*, which puts each node's error in return units.

Lemma 2 (Units reduction). *Let a node's* clean-input *intrinsic error be $\varepsilon_j := d^{(j)}_{\mathcal{A}}(a^{\mathrm{real}}_j, a^{\mathrm{ideal}}_j)$ on* ideal helper inputs. *Then $\varepsilon_j \le \Delta^{\mathrm{r}}_j + \Delta^{\mathrm{e}}_j + \Delta^{\mathrm{tool}}_j + \Delta^{\mathrm{int}}_j$, so ε_j is in realizable-return units and the sink error $\delta_{\mathrm{out}}(t)$ is a realizable-return gap. The helper-input rung Δ^{help}_j is excluded from ε_j by construction; it re-enters through propagation.*

Remark 1 (Helper input is the propagated term). Δ^{help}_j is exactly the return lost because node j's parents deliver imperfect artifacts, the input deviation $\delta_{\mathrm{in}}(j) = \sum_{i \in \mathrm{pa}(j)} \delta_{\mathrm{out}}(i)$, which enters as the term $L_j\, \delta_{\mathrm{in}}(j)$ of the propagation recurrence (Theorem 1). So Δ^{help} is not dropped: it is the cross-node coupling, accounted for once, by propagation. This is what makes the exact identity (Lemma 1) and the propagation bound consistent: ε_j carries the node-local rungs, Δ^{help} the inherited one.

Lemma 2 is the pivot of the paper: every node-local imperfection becomes one number ε_j in the certificate's units, and the inherited error routes into propagation, which the next section feeds through the composition DAG. (Proofs: Appendix B.)

5 Composition, Propagation, and Scheduling

We bound the *composition* term of (1). The input is the per-node ε_j from Lemma 2; the output is $\mathbb{E}[\Delta^{\mathrm{comp}}] \le B(t) + P^{\mathrm{rup}}(t)$ with the checkpoint design rule and the measured constants (L_j, p_j) that make it usable.

Theorem 1 (DAG propagation under ruptures). *Write $i \preceq t$ if there is a directed path $i \rightsquigarrow t$ (or $i = t$); let the path amplification be $\Lambda(p) = \prod_{(u\to v)\in p} L_{u\to v}$; define the* stable-manifold bound *and* rupture budget

$$B(t) := \min\Big(1,\ \sum_{i \preceq t} \varepsilon_i \sum_{p:\, i \rightsquigarrow t} \Lambda(p)\Big), \qquad P^{\mathrm{rup}}(t) := \sum_{j \preceq t} p_j.$$

Then $\mathbb{P}[\delta_{\mathrm{out}}(t) \le B(t)] \ge 1 - P^{\mathrm{rup}}(t)$ (high probability) and $\mathbb{E}[\delta_{\mathrm{out}}(t)] \le B(t) + P^{\mathrm{rup}}(t)$ (in expectation), and $\mathbb{E}[\Delta^{\mathrm{comp}}(\mathcal{P})] \le \mathbb{E}[\delta_{\mathrm{out}}(t)]$ for the plan composition gap $\Delta^{\mathrm{comp}}(\mathcal{P}) := J^\circ_\tau - J_\tau(\mathcal{P})$.

(Proof in Appendix B.1.) On the joint stable event every ancestor obeys its local bound; unrolling gives $\delta_{\mathrm{out}}(t) \le B(t)$, a union bound gives $\mathbb{P}[S^c] \le P^{\mathrm{rup}}(t)$, and the trivial bound on S^c gives the expectation. This is the technical core of (1): contraction controls amplification, ruptures are a separate additive budget. Two corollaries make it actionable.

Corollary 1 (Depth trichotomy with rupture accumulation). *For a depth-d chain with uniform (L, ε, p), $\mathbb{E}[\delta_{\mathrm{out}}(d)] \le B(d) + dp$ with stable term $B(d) = \min(1, \varepsilon(L^d - 1)/(L - 1))$ ($L \ne 1$) or $\min(1, d\varepsilon)$ ($L = 1$). Thus $L < 1$ gives a depth-independent $B \le \varepsilon/(1 - L)$ (contractive); $L = 1$ gives linear $B = d\varepsilon$ (the additive penalty of prior work); $L > 1$ gives exponential $B = \Theta(\varepsilon L^d)$ (the snowball regime Gan et al. (2025); Cemri et al. (2025)). In every regime depth costs an extra $\Theta(dp)$ through ruptures: contraction does not suppress rupture accumulation; only checkpointing does.*

Corollary 2 (Checkpoints suppress amplification and ruptures). *A checkpoint with contraction $L_v < 1$ and rupture-catch probability q inserted every k steps of an $L > 1$ chain caps per-segment amplification at $L^k L_v$, so if $L^k L_v < 1$ the stable error is depth-independent, and it scales the rupture budget to $P^{\mathrm{rup}} \le (1 - q)\, dp$ (vanishing as $q \to 1$). The "every k steps" heuristic is replaced by optimal placement of a checkpoint budget on a general DAG (Algorithm 2, Appendix C).*

Matching lower bounds (Thm 5, App. B). Both terms are order-*tight*: a rupture-free chain attains $B = \varepsilon(L^d - 1)/(L - 1)$ with equality (no $o(\varepsilon L^d)$ bound holds for $L > 1$), and an amplification-free chain with independent rupture p attains $\mathbb{E}[\delta_{\mathrm{out}}(d)] = \Theta(dp)$ (no $o(dp)$), so $\mathcal{C} = \min(1, B) + P^{\mathrm{rup}}$

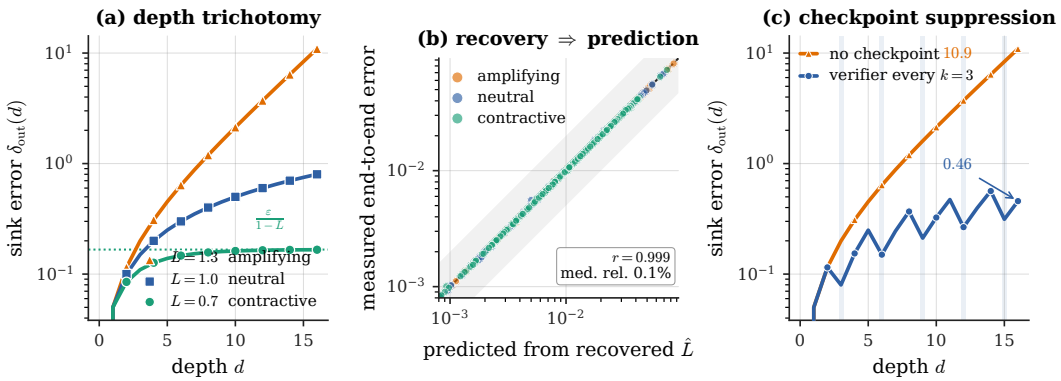


Figure 2: Propagation (validation suite; intrinsic error $\varepsilon = 0.05$). **(a)** Depth trichotomy (Corollary 1): contractive saturates, neutral is linear, amplifying is exponential; markers are simulation, lines theory, coinciding. **(b)** *Recovery and prediction*: across 600 random black-box pipelines of depth 2–9, end-to-end error predicted from *perturbation-recovered* local gains $\hat{L}_j$ matches measured error (log-correlation $r = 0.999$, median relative error 0.15%), evidence the law predicts out of sample rather than restating an identity. **(c)** A verifier ($L_v = 0.4$) every $k = 3$ steps caps an explosive $L = 1.3$ chain (Corollary 2).

is the *best possible* bound of its form: amplification and rupture are separately unavoidable. An average-case companion (Prop 7) puts the non-vacuity threshold at the sign of the Lyapunov exponent $\lambda = \mathbb{E}[\log L]$, not the maximum; across parallel branches, errors and rupture budgets add (Cor 3).

Recovering the constants by perturbation. Both governing quantities are measurable on a real skill without opening it: perturb its input artifact, sample the output, and record the drift; a perturbation is a *rupture* when the output is disproportionate (crosses a categorical boundary, i.e. refusal, invalid format, or schema break, or exceeds a high drift quantile). Then $\hat{p}_j$ is the rupture rate and $\hat{L}_j$ the robust (Theil–Sen) slope on the non-rupture set; plugging $(\hat{L}_j, \hat{p}_j)$ into Theorem 1 predicts end-to-end error. The bound holds for every rupture threshold and we report the tightest (Proposition 6); a Lipschitz-in-proxy regularity makes the cheap proxy metric a conservative input (Proposition 5); and certifying p_j to accuracy η costs $\Theta(\eta^{-2} \log \delta^{-1})$ queries (Proposition 8), all in Appendix B. Library-scale allocation under a query budget is Algorithm 4.

Scheduling (timing). Composition has a second face: when reward arrives. With w independent subtasks of latency ℓ_i^{lat} and reward R_i, a parallel schedule earns $\sum_i \gamma^{\ell_i^{\text{lat}}} R_i$ and any serialization earns $\sum_i \gamma^{t_i} R_i$ with $t_i = \sum_{j \sqsubseteq i} \ell_j^{\text{lat}} \geq \ell_i^{\text{lat}}$; the scheduling gap is $\Delta^{\text{sched}} = \sum_i R_i(\gamma^{\ell_i^{\text{lat}}} - \gamma^{t_i}) \geq 0$, which is $\Theta(w)$ in discounted return and $\to 0$ as $\gamma \to 1$ (Theorem 6, Appendix B). It enters the certificate additively, $J_\tau^\circ - \mathbb{E}[J_\tau(\mathcal{P})] = [\text{composition}] + \Delta^{\text{sched}}$ (Cor. 6), so timing obeys the same inequality (latency ℓ^{lat} is distinct from the length budget ℓ).

6 Library Size and Learnability

Section 5 bounded composition given the per-skill errors; here we bound those at their source. A library is ε-*sufficient* if every supported task has a retrievable skill with realizable gap $\leq \varepsilon$.

Theorem 2 (Library size: upper and lower bounds)**.** *Under **A1–A3**: (i) some library of $M(\varepsilon/2)$ skills is $(\varepsilon + \varepsilon_0)$-sufficient, so the retrieval term obeys $\Delta^{\text{r}} \leq \varepsilon_{\text{cov}}$ once $\mathcal{L}$ is a cover; (ii) any ε-sufficient library has size $\geq \mathcal{N}_\varepsilon^{\text{sep}}(\mathcal{T})$, the largest pairwise-$\varepsilon$-separated task set under joint sub-optimality ρ. Authoring an ε-excess skill from n i.i.d. instances needs $n = O(\varepsilon^{-2} \log(|\mathcal{B}|/\delta))$ (Theorem 7), for total data $\widetilde{O}(M(\varepsilon/2)\, \varepsilon^{-2} \log(|\mathcal{B}|/\delta))$.*

The upper bound is a covering number under $d_{\mathcal{T}}$, the lower a separation number under ρ; Example 1 (Appendix B) computes both as $\Theta(\varepsilon^{-n})$ for an explicit family, tight up to constants: minimal library size is exponential in the intrinsic task dimension. (Transfer Lemma 4 and proofs: Appendix B.)

7 Safety Under Evolution

Everything so far holds for a fixed library; (1) still needs its horizon $T^\star$. Libraries are edited continually, including granting tools; self-evolving agents empirically lose safety Shao et al. (2025). We track the maximum danger $P_t := \max_{\sigma \in S_t} p(\sigma)$, where $p(\sigma)$ is the probability of a forbidden action under σ. Edits split into *continuous* operations (rewrite/retrigger/refactor; move P_t by $\le b$) and *jumps* (add/grant/merge; move P_t by $\le \beta$, at most k times).

Theorem 3 (Two-regime safety)**.** *With $P_0 \le p^0$ and at most k jumps of size $\le \beta$, the per-step change $\xi_t = P_t - P_{t-1}$ satisfies:* (adversarial, $|\xi_t| \le b$) $P_t \le p^0 + bt + \beta k$*; and* (benign, additionally $\mathbb{E}[\xi_t \mid \mathcal{F}_{t-1}] \le \mu$) *with probability $\ge 1-\delta$,*

$$P_t \le p^0 + \mu t + 2b\sqrt{2t\ln(1/\delta)} + \beta k.$$

(Proof in Appendix B.) The proof centers the increments into a martingale-difference sequence (valid although P_t is a *submartingale* maximum) and applies Azuma–Hoeffding. The adversarial bound is the right guarantee when an optimizer chooses edits; the benign envelope is far tighter and yields the deployment window $T^* = \max\{t : p^0 + \mu t + 2b\sqrt{2t\ln(1/\delta)} + \beta k \le p_{\max}\}$ (Corollary 4); their difference is the safety value of benign evolution. A shared workspace couples skills through artifact taint, inflating b, μ by $(1+\chi)$ (Proposition 9, Appendix B).

Compositional danger: tools in combination. Single-skill bounds miss the defining risk of artifact-sharing agents: a plan of individually-safe skills can be jointly unsafe. Let each skill have capabilities $\mathrm{cap}(\sigma) \subseteq C$ (read-private, net-egress, exec, …); on rupture it may exercise $\le c$ extra capabilities (c a runtime-enforceable per-action budget). Dangerous conjunctions $\mathcal{D}$ are capability sets unsafe along one flow (e.g. $\{$read-private, net-egress$\}$ = exfiltration); along a flow p they accumulate, $\mathrm{cap}(p) = \bigcup_{\sigma \in p} \mathrm{cap}(\sigma)$.

Proposition 1 (Single-skill danger does not bound compositional danger)**.** *There is a plan with $\max_\sigma p(\sigma) = 0$ yet compositional danger $P^{\mathrm{comp}}(\mathcal{P}) = 1$.*

The witness: σ_1 reads private data, σ_2 has net-egress; individually safe, but the flow $\sigma_1 \to \sigma_2$ exfiltrates with certainty. Hence Theorem 3 (which bounds $\max_\sigma p$) says nothing about compositional danger; a separate criterion is required. Because LLM ruptures across stages are *correlated* (shared base model), we use the worst-case *conditional* rupture rate $\bar{p} := \max_\sigma \max_B \mathbb{P}[R_\sigma \mid \bigcap_{\sigma' \in B} R_{\sigma'}]$ (under independence $\bar{p} = p_{\max}$; under positive correlation $\bar{p} \ge p_{\max}$, correctly weakening the guarantee).

Theorem 4 (Capability-flow safety)**.** *A plan has* margin k *if $|d \setminus \mathrm{cap}(p)| \ge k$ for every flow p and every $d \in \mathcal{D}$. Under per-action budget c, realizing any d needs $\ge \lceil k/c \rceil$ skills on a flow to rupture, and*

$$\begin{aligned} P^{\mathrm{comp}}(\mathcal{P}) \;&\le\; \sum_{d \in \mathcal{D}} \sum_{\text{flows } p} N_p(d)\, \bar{p}^{\,\lceil k/c \rceil} \\ &= O\big(\mathrm{poly}(|\mathcal{P}|) \cdot \bar{p}^{\,\lceil k/c \rceil}\big). \end{aligned}$$

A margin-k library thus demotes compositional danger to order $\lceil k/c \rceil$ in $\bar{p}$, genuinely higher-order iff the budget is capped ($k > c$) and ruptures are decorrelated enough ($\bar{p} < 1$).

(Proof in Appendix B.3.) The triple $(k, c, \bar{p})$ is checkable: (k, c) from the static assignment and sandbox, $\bar{p}$ from paired-rupture measurement, so it enters the certificate alongside the other designed-down terms; a margin-k, budget-c, checkpointed library carries danger $O((1-q)^{\lceil k/c \rceil} \bar{p}^{\lceil k/c \rceil})$. The margin drifts under capability grants, giving a second horizon $T^\star_{\mathrm{comp}}$ (Corollary 5, Appendix B), so the certificate's window is $\min(T^\star, T^\star_{\mathrm{comp}})$, the static design-time counterpart of runtime trusted-action enforcement Costa et al. (2025).

8 SYNTHESIS: THE AGENT CERTIFICATE

The framework fixed the target; the decomposition named its terms and reduced each skill's error to ε_j; propagation turned the ε_j into a bounded composition term with measurable constants; size and authoring bounded the retrieval term and the inputs to ε_j; safety supplied the horizon. Substituting yields (1).

Proposition 2 (Agent certificate)**.** *Fix a task τ and a plan $\mathcal{P}$ with sink t. Let $\Lambda_i^{\to t} = \sum_{p:\, i \rightsquigarrow t} \Lambda(p)$ be node i's total amplification to the sink and $\varepsilon_i \le \Delta_i^{\mathrm{r}} + \Delta_i^{\mathrm{e}} + \Delta_i^{\mathrm{tool}} + \Delta_i^{\mathrm{int}} =: g_i$ (Lemma 2). Then*

$$J_\tau^\circ - \mathbb{E}[J_\tau(\mathcal{P})] \;\le\; \min\Big(1,\, \sum_{i \preceq t} g_i\, \Lambda_i^{\to t}\Big) \;+\; P^{\mathrm{rup}}(t), \tag{3}$$

and every quantity on the right is controllable or measurable: $\Delta_i^{\mathrm{r}} \le \varepsilon_{\mathrm{cov}}$ once $\mathcal{L}$ covers (Theorem 2); $\Delta_i^{\mathrm{e}} = O(\sqrt{\log(\cdot)/m})$ in the number m of authoring instances (Theorem 7); $\Delta_i^{\mathrm{int}} \le \nu$ (Proposition 4); the helper rung is already inside $\Lambda_i^{\to t}$ (Remark 1); and $\Lambda, P^{\mathrm{rup}}$ are recovered black-box, with checkpoints making the first term depth-independent and scaling P^{rup} by $(1-q)$ (Corollary 2). The bound applies for $t \le \min(T^\star, T^\star_{\mathrm{comp}})$, where the benign envelope keeps single-skill danger within budget and the capability margin keeps compositional danger at order $\lceil k/c \rceil$ (Theorems 3, 4).

Reading the certificate. Equation (3) is a *design target*: shrink Δ^{r} by coverage, Δ^{e} by authoring, $\Delta^{\mathrm{tool}}, \Delta^{\mathrm{int}}$ by tool quality, and, most importantly for depth, keep the amplification product and rupture budget small by measuring (L_j, p_j) and inserting checkpoints. But the terms are additive only in accounting, not under *intervention* (editing one moves another, via the intervention-coupling bound, Proposition 10), so (3) is optimized jointly (Algorithm 6), not term by term.

9 ALGORITHMS: MAKING THE CERTIFICATE CONSTRUCTIVE

The analysis names the quantities in (3); five algorithms construct and optimize them, each with a guarantee. The certificate *reduction* $f(\cdot) = \mathcal{C}(\emptyset) - \mathcal{C}(\cdot)$ is monotone *submodular* in the design set (coverage-type functions; substitutive cross-effects preserve submodularity, as a checkpoint lowering Λ_j also lowers the value of authoring j), so cost-benefit greedy attains the stated ratios for checkpoint placement and joint optimization. GATEFORMARGIN (Algorithm 3) is min-cut for pairwise conjunctions and submodular set cover in general (NP-hard, $\ln$-approximable); NEYMANALLOCATE (Algorithm 4) uses $n_i \propto \Lambda_i \sqrt{p_i(1-p_i)}$; and BEHAVIORALRERANK (Algorithm 5) re-ranks an embedding top-K by measured return, neutralizing embedding distortion whenever recall@K holds. Appendix C gives full pseudocode, proofs, the intervention-coupling analysis, and a one-line summary of all five with their guarantees (Table 3).

10 EXPERIMENTS

Two findings matter most. First, the certificate's deploy/reject verdict *discriminates*: it separates deployable from non-deployable pipelines across both models and data domains, not firing uniformly. Second, on real end-to-end LLM pipelines the measured loss stays *within* the bound in every one of the 27 cells. Beneath them, the chain validates piece by piece with no result hand-set: the decomposition holds to machine precision, size and authoring laws scale as predicted, and the safety envelope contains all trajectories (full matrix in Table 4, Appendix D; the released suite machine-checks 32 theory results and 5 algorithms).

Propagation recovers and predicts (Fig. 2). Panel (a) reproduces the trichotomy (theory and simulation coincide). Panel (b) builds 600 random black-box pipelines, *recovers* each stage's sensitivity $\hat{L}_j$ by central-difference perturbation *without knowing the stage*, and predicts end-to-end error via $B(t)$ matching measured error (log-correlation 0.999, median relative error 0.15%). Panel (c) confirms checkpoint stabilization (error at depth 16: $10.9 \to 0.46$). On a planted non-Lipschitz model the estimator recovers sensitivities within 3% and rupture rates within 0.07 of planted values, $B(t) + P^{\mathrm{rup}}(t)$ holding out of sample.

Table 1: Certificate grid (live; six-database suite): useful depth $d^\star$ (bare $\rightarrow$ with one checkpoint k=2, L_v=0.4, q=0.9) per (model, domain), the deepest pipeline still certifiable ($\mathcal{C} \leq \theta$=0.5); deployable when $d^\star \geq 5$. The reasoning families (‡) stay non-deployable on code/math. Mean 922 live calls/model.

Model	text	data	code	math	fin	web
Llama-3.1-8B	$2\rightarrow7$	$10\rightarrow10$	$1\rightarrow5$	$3\rightarrow8$	$2\rightarrow7$	$3\rightarrow8$
Llama-3.3-70B	$0\rightarrow5$	$2\rightarrow7$	$1\rightarrow5$	$1\rightarrow6$	$1\rightarrow5$	$3\rightarrow8$
Llama-4-Mav-17B	$3\rightarrow8$	$1\rightarrow10$	$0\rightarrow4$	$1\rightarrow5$	$1\rightarrow6$	$1\rightarrow5$
gpt-oss-20B‡	$0\rightarrow5$	$0\rightarrow4$	$0\rightarrow3$	$0\rightarrow3$	$0\rightarrow3$	$1\rightarrow5$
Qwen3-80B‡	$0\rightarrow5$	$1\rightarrow16$	$0\rightarrow3$	$0\rightarrow3$	$0\rightarrow4$	$2\rightarrow7$

Real skill libraries (Table 5, Appendix D). We instantiate the framework on a *six-database,* 332*-skill* suite (text/data/code/math/finance/web-security), each a runnable `SKILL.md` with a tool and bundled artifacts, plus three independent public SKILL.md libraries for external validity (359 skills, official + community; Appendix G). Measured per-skill sensitivities span all three regimes of Corollary 1 (247 contractive, 49 neutral, 36 amplifying), and both rupture channels fire: strict structured-data tools rupture on malformed input ($\hat{p}^{\text{tool}} = 1.0$ on data/math), lenient text transforms do not (0.0). Lexical retrieval over the 664 triggers gives recall@1 = 0.66, recall@3 = 0.80, bounding Δ^{r}; end-to-end pipeline propagation ($B(t) + P^{\text{rup}}(t)$ upper-bounding measured error) is in Table 6.

Compositional safety (Table 7, Appendix D). Across the suite no single skill carries a dangerous capability pair, yet dangerous conjunctions form *across* skills (read-private $\times$11, write-public $\times$9, net-egress $\times$6, exec $\times$5): a private-reader composed with a public-writer is individually safe but jointly exfiltrates, the separation of Proposition 1 made concrete. A capability margin k demotes joint danger to order $\lceil k/c \rceil$ (Theorem 4); with a checkpoint the corrected margin-2 danger drops from a vacuous 1.0 to 0.006 (Table 11). Two-regime safety holds over 250 benign trajectories, $0/250$ envelope violations (Fig. 3, Appendix D).

A live certificate grid: five models $\times$ six domains (Table 1). We measure the LLM channel *live* for five models from three families (Table 1) over six domains (4,612 calls). Each cell yields $(\hat{L}, \hat{p})$, the certificate, and its useful depth $d^\star$ (Thm. 8, deployable when $d^\star \geq 5$). The verdict *discriminates on both axes*: only $1/30$ cells is deployable bare, one checkpoint rescues $22/30$ (at *assumed* recall $q = 0.9$; measured-verifier reach is thinner, App. G), and the reasoning families stay non-deployable on code and math, a structured failure the certificate *localizes*. The rollup, ground-truth anchoring, strict-tool cascade, external validity, and seven further theorems are in Appendix G.

End to end on real LLM pipelines (Table 8). We compose actual skills into *live* depth-d pipelines across a *nine-model, six-family sweep*: measured loss stays within the certificate in *all* 27 *cells* (Table 8 shows five). *Two* models certify DEPLOY on the benign pipeline (Llama and the modern Mistral-Small), and a *real* verifier (q, f measured, not the assumed 0.9) flips Llama's deep-pipeline verdict to DEPLOY ($\mathcal{C} = 0.42$, 4$\times$ loss cut); this flip is Llama's alone (1 of 9), needing near-neutral amplification *and* in-loop-detectable corruption that stronger models launder past the verifier. Rejects localize distinct causes: high loss, high rupture, saturated amplification (Mixtral, a safe-side false negative; Appendix G).

11 Discussion, Limitations, and Conclusion

We produced one object: a certificate (3) bounding the return a composed agent loses against the best its frozen model can achieve, every term controllable or measurable and valid over a horizon that survives library edits. Stated against the realizable optimum for budget ℓ, it is intractable to compute but conservatively bounded through measurable proxy constants (Propositions 3, 5). *Open*: the safety bounds assume correct μ, b, β, k and benign editing; implicit information flows (through control, not data) are uncovered; learning the plan decomposition and capability assignment with guarantees remains open; and the live e2e sweep is one seed, verifier, and vendor. This gives designers a single inequality to drive down, each knob attached to a term it provably controls.

ETHICS STATEMENT

This theoretical paper's safety results certify and constrain tool-augmented agents rather than enable misuse: the compositional-danger separation (Proposition 1) uses an abstract exfiltration example with no operational detail, and the harness measures only input–output sensitivity. By making an agent's loss and danger measurable, the work aims to make deployed skill-library systems safer and more auditable.

AI USE STATEMENT

Per the ICLR 2027 AI Policy for Authors, large language models (LLMs) were used solely to aid and polish writing (grammar, phrasing, and clarity). No AI tool contributed to the research ideas, theorems, proofs, algorithms, experiments, or analysis, all of which are the authors' own; all AI-assisted text was reviewed and verified by the authors, who take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. All assumptions (**A1**–**A3**, Assumption 1) are stated in Section 3, and complete proofs of every theoretical claim (Lemmas 1–2, Theorems 1, 2, 3, 4, their corollaries, and all deferred results) are given in Appendix B, with the notation glossary in Appendix A and a theory reproducibility checklist in Appendix E. Pseudocode and guarantees for the five algorithms of Section 9 are in Appendix C. The experimental protocol, the full validation matrix, the six-database 332-skill suite, and the live model$\times$domain grid are described in Appendix D, with additional experiments and external-validity results on public SKILL.md libraries in Appendix G; the released harness machine-checks all 32 theory results and 5 algorithms.

REFERENCES

Anthropic. Introducing the Model Context Protocol (MCP). https://www.anthropic.com/news/model-context-protocol, 2024. Accessed: 2026-07-13.

Anthropic. Equipping agents for the real world with agent skills (SKILL.md). https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills, 2025. Accessed: 2026-07-13.

K. Asadi, D. Misra, and M. L. Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

K. Azuma. Weighted sums of certain dependent random variables. *Tôhoku Mathematical Journal*, 19(3):357–367, 1967. doi: 10.2748/tmj/1178243286.

Y. Bai, F. Chen, H. Wang, C. Xiong, and S. Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341–379, 2003. doi: 10.1023/A:1025696116075.

P. S. Castro, T. Kastner, P. Panangaden, and M. Rowland. MICo: Improved representations via sampling-based state similarity for Markov decision processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

M. Cemri, M. Z. Pan, S. Yang, L. A. Agrawal, et al. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

A. Chen, J. Phang, A. Parrish, et al. Two failures of self-consistency in the multi-step reasoning of LLMs. *Transactions on Machine Learning Research (TMLR)*, 2024.

K. Cobbe et al. Training verifiers to solve math word problems, 2021.

F. Cole, Y. Zhao, Y. Lu, and T. Zhang. In-context learning of linear dynamical systems with transformers: Approximation bounds and depth-separation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

M. Costa, B. Köpf, A. Kolluri, A. Paverd, M. Russinovich, et al. Securing AI agents with information-flow control (Fides), 2025.

E. Debenedetti, J. Zhang, M. Balunović, et al. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

E. Debenedetti et al. Defeating prompt injections by design (CaMeL), 2025.

D. E. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5): 236–243, 1976. doi: 10.1145/360051.360056.

J. B. Dennis and E. C. Van Horn. Programming semantics for multiprogrammed computations. *Communications of the ACM*, 9(3):143–155, 1966. doi: 10.1145/365230.365252.

T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research (JAIR)*, 13:227–303, 2000. doi: 10.1613/jair.639.

J. Duan, J. Diffenderfer, S. Madireddy, T. Chen, B. Kailkhura, and K. Xu. UProp: Investigating the uncertainty propagation of LLMs in multi-step agentic decision-making, 2025.

N. Dziri et al. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

K. Ellis, C. Wong, M. Nye, et al. DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*, 2021. doi: 10.1145/3453483.3454080.

N. Ferns, P. Panangaden, and D. Precup. Bisimulation metrics for continuous Markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011. doi: 10.1137/10080484X.

H. Furstenberg and H. Kesten. Products of random matrices. *The Annals of Mathematical Statistics*, 31(2):457–469, 1960. doi: 10.1214/aoms/1177705909.

Z. Gan, Y. Liao, and Y. Liu. Rethinking external slow-thinking: From snowball errors to probability of correct reasoning. In *International Conference on Machine Learning (ICML)*, 2025.

H.-a. Gao et al. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research (TMLR)*, 2026.

S. Garg, D. Tsipras, P. Liang, and G. Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec)*, 2023. doi: 10.1145/3605764.3623985.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association (JASA)*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

A. Hosseini, A. Vani, D. Bahdanau, A. Sordoni, and A. Courville. On the compositional generalization gap of in-context learning. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2022. doi: 10.18653/v1/2022.blackboxnlp-1.22.

J. Huang et al. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024a.

J.-t. Huang, J. Zhou, T. Jin, X. Zhou, Z. Chen, W. Wang, Y. Yuan, M. R. Lyu, and M. Sap. On the resilience of LLM-based multi-agent collaboration with faulty agents, 2024b.

M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Y. Jiang, D. Li, H. Deng, B. Ma, X. Wang, Q. Wang, and G. Yu. SoK: Agentic skills: Beyond tool use in LLM agents, 2026.

R. Kamoi, Y. Zhang, N. Zhang, et al. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics (TACL)*, 12:1417–1440, 2024. doi: 10.1162/tacl_a_00713.

M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994. doi: 10.1007/BF00993468.

O. Khattab et al. DSPy: Compiling declarative language model calls into self-improving pipelines. In *International Conference on Learning Representations (ICLR)*, 2024.

S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999. doi: 10.1016/S0020-0190(99)00031-9.

A. Krause and C. Guestrin. A note on the budgeted maximization of submodular functions. Technical Report CMU-CALD-05-103, Carnegie Mellon University, 2005.

B. M. Lake and M. Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning (ICML)*, 2018.

P. Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

F. Li, P. Tagkopoulos, and I. Tagkopoulos. SkillFlow: Scalable and efficient agent skill retrieval system, 2025.

X. Li et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks, 2026.

Y. Li, M. E. Ildiz, D. Papailiopoulos, and S. Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning (ICML)*, 2023.

H. Lightman et al. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

A. Madaan et al. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions, I. *Mathematical Programming*, 14(1):265–294, 1978. doi: 10.1007/BF01588971.

J. Neyman. On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4): 558–606, 1934. doi: 10.2307/2342192.

E. Perez et al. Red teaming language models with language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. doi: 10.18653/v1/2022.emnlp-main.225.

O. Press et al. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics (EMNLP)*, 2023. doi: 10.18653/v1/2023.findings-emnlp.378.

S. Ross and J. A. Bagnell. Efficient reductions for imitation learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

S. Ross, G. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

A. Sabelfeld and A. C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003. doi: 10.1109/JSAC.2002.806121.

Y. Sakai, H. Kamigaito, and T. Watanabe. Revisiting compositional generalization capability of LLMs considering instruction-following ability. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2025.

T. Schick et al. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

P. K. Sen. Estimates of the regression coefficient based on kendall's tau. *Journal of the American Statistical Association (JASA)*, 63(324):1379–1389, 1968. doi: 10.1080/01621459.1968.10480934.

S. Shao, Q. Ren, C. Qian, et al. Your agent may misevolve: Emergent risks in self-evolving LLM agents, 2025.

M. Sharma, M. Tong, J. Mu, et al. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming, 2025.

Z. Shi et al. Retrieval models aren't tool-savvy: Benchmarking tool retrieval for large language models. In *Findings of the Association for Computational Linguistics (ACL)*, 2025.

N. Shinn et al. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1.

M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004. doi: 10.1016/S0167-6377(03)00062-2.

H. Theil. A rank-invariant method of linear and polynomial regression analysis. *Nederl. Akad. Wetensch. Proc.*, 53:386–392, 521–525, 1397–1412, 1950.

J. Uesato et al. Solving math word problems with process- and outcome-based feedback, 2022.

S. Vijayvargiya, A. B. Soni, X. Zhou, et al. OpenAgentSafety: A comprehensive framework for evaluating real-world AI agent safety, 2025.

M. J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.

G. Wang et al. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

L. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982. doi: 10.1007/BF02579435.

S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022.

S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

S. Yao et al. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

M. Zaharia, O. Khattab, L. Chen, et al. The shift from models to compound AI systems. https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/, 2024. Accessed: 2026-07-13.