COMPARING MODELS, SKILLS, FEEDBACK AND CURRICULUM DESIGN FOR AGENTIC ROBOTICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic robotics asks coding agents to implement and iteratively improve robot policies from simulated and real-world physical interactions. We study the self-learning loop of agentic robotics through a controlled simulation study spanning nine manipulation tasks, 631 different experiment configurations, and 28K coding agent rollouts, systematically varying model capability, skill priors, feedback structure, and curriculum. The experiment data suggests that the bottleneck is often not the access to sufficient information, but interpreting it into the right policy update. Across these factors, a successful self-learning loop depends on how effectively agents attribute failures, reason task-relevant spatial and geometric structure, and selectively restructure code to improve policies.

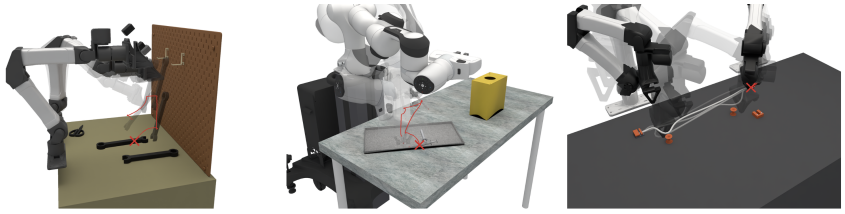

Figure 1: **Representative failure modes in Agentic Robotics.** Left: *Manipulation planning* fails to select a grasp and approach strategy compatible with the downstream placement. Center: *Motion planning* fails to jointly reason about held-object geometry, obstacles, and clearance to produce a collision-free trajectory. Right: *Deformable manipulation* requires reasoning over object configuration and contact that is difficult to capture through local motion decisions.

1 INTRODUCTION

Recent coding agents, such as GPT Astra (OpenAI, 2026a) and Claude Fable (Anthropic, 2026a), have become increasingly capable of generating and iteratively repairing executable programs, motivating a growing class of Agentic Robotics (AR) systems in which LLM- and VLM-based agents write code that composes perception, planning, and control skills into robot policies (Liang et al., 2023b; Singh et al., 2023; Fu et al., 2026; Chen et al., 2026). However, self-improvement in robotics differs fundamentally from software debugging. In software, failures, such as compiler errors, are often exposed through symbolic feedback that is already closely aligned with the program; but in robotics, the evidence to improve a policy is instead embedded in physical interactions: object geometry, spatial relationships, contact, motion, and changes in the scene over time. As illustrated in Figure 1, a failed placement can be attributed to an earlier incompatible grasp and a collision requires reasoning about the geometry of an object after it becomes part of the moving robot. In this paper, we ask: what determines whether an agent can interpret physical interaction and convert it into effective robot-policy improvement?

We design a controlled simulation study of agentic robot self-learning across five contact-rich VABAR manipulation tasks Xie et al. (2026b) and four longer-horizon MimicGen tasks Mandlekar et al. (2023). Holding the overall self-learning harness fixed, we systematically vary four components of the learning loop: model capability, skill prior, feedback structure, and curriculum, as illustrated in Figure 2. Across these studies, we find that self-learning is limited less by the availability of useful information than by the agent's ability to turn the information into effective policy

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.

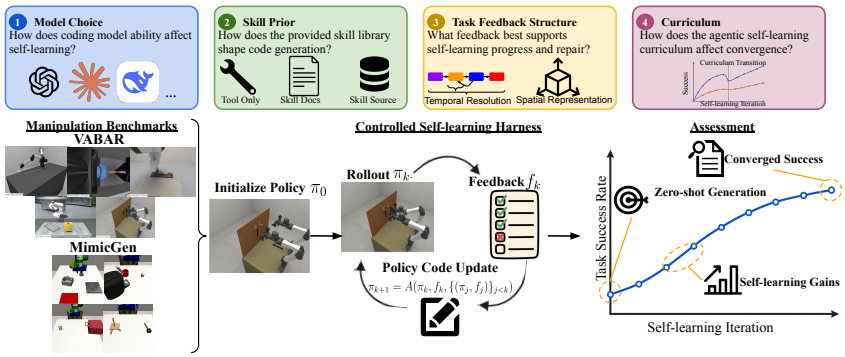

Figure 2: Overview of our controlled study of agentic robot self-learning of model, skill, feedback, and curriculum design. A coding agent initializes a policy π_0, executes the current policy π_k in simulation, receives feedback f_k, and iteratively edits its policy as $\pi_{k+1} = A(\pi_k, f_k, \{(\pi_j, f_j)\}_{j<k})$. We evaluate zero-shot and self-learning performance, and final task success across VABAR and MimicGen manipulation tasks.

updates. More execution feedback is not necessarily more useful: feedback is most effective when aligned with semantically meaningful policy decisions rather than reported at the finest temporal resolution. Similarly, world-frame states contain all the information needed for manipulation, yet agents often fail to recover task-relevant object-relative and functional relationships unless these are made explicit. Even when agents identify local geometric errors, they often struggle to translate them into coherent collision-aware motion plans. We find that this limitation can also affect curriculum design: agents tend to preserve and patch policy structure inherited from earlier stages rather than fundamentally reorganize it, so privileged learning can either provide a useful initialization or constrain later learning when the inherited policy depends on privileged precision or brittle manipulation strategies. Our results further suggest that progress in agentic robot self-learning comes from the better structures of skills, feedback, and state representations and the learned physical reasoning ability from the underlying models.

This paper makes three contributions: (1) A controlled study design for Agentic Robotics that isolates four important components of simulation-guided self-learning, (2) Experiment data from 631 implemented experiments with 28 thousand simulation rollouts on Agentic robotics policies, (3) Six findings on model capability, skill priors, feedback structure, and curriculum.

2 Related Work

Agentic Robotics. Foundation models have been used to compose robot skills and generate executable policies through language-model planning and code generation (Liang et al., 2023a; Ahn et al., 2023; Singh et al., 2023). More recent work integrates foundation models with explicit spatial representations, geometric constraints, and task-and-motion planning to support open-vocabulary and multi-step manipulation (Huang et al., 2023a; 2024b; Shen et al., 2026; Murray et al., 2025). Other approaches close the loop by using simulation or real-world execution feedback to revise policies, acquire skills, and improve task performance over repeated interactions (Fu et al., 2026; Chen et al., 2026; Xiao et al., 2026; Lu et al., 2026). Rather than introducing a new agent architecture, we study how individual components of this self-learning loop affect performance.

Controlled Evaluation of Agentic Robotics. Robot-learning benchmarks increasingly use standardized task suites to evaluate generalization, long-horizon reasoning, and manipulation under controlled variation (Gu et al., 2023; Li et al., 2022; Mees et al., 2022; Liu et al., 2023; Zheng et al., 2022). More recent benchmarks extend controlled evaluation to foundation-model agents by measuring interactive decision making, embodied reasoning, perception, and action across diverse environments (Liu et al., 2024; Manling Li et al., 2024; Rui Yang et al., 2025). Robot-specific evaluations further study robot-centric perception and decision making, coding-agent behavior under different abstraction and interaction settings, and repeated manipulation under controlled task variation (Xie et al., 2026a; Fu et al., 2026; Xie et al., 2026b). Unlike prior agentic robotics benchmarks that primarily vary tasks, environments, or complete agent systems, we hold the self-learning har-

ness fixed and independently vary model capability, skill priors, feedback structure, and curriculum to isolate how each factor affects initial policy generation and subsequent self-improvement.

Skills, Feedback, and Curriculum for Self-Learning. Prior work has shown that reusable skills and execution feedback can improve embodied agent learning (Ahn et al., 2023; Wang et al., 2024; Huang et al., 2023b; Fu et al., 2026). Skill libraries provide reusable action abstractions that constrain generation and support compositional behavior, while systems such as Voyager jointly combine an executable skill library, environment feedback, and an automatic curriculum for continual improvement (Wang et al., 2024). More broadly, language agents can use environment observations, linguistic critiques, or tool-derived feedback to revise plans and outputs across interaction or repeated attempts (Yao et al., 2023; Shinn et al., 2023; Madaan et al., 2023; Gou et al., 2024). However, prior work also shows that self-correction can be unreliable without sufficiently informative external feedback (Huang et al., 2024a). Curriculum learning instead structures the progression of experience (Bengio et al., 2009), while privileged-information methods use state available during training to simplify learning before transferring to image-based or otherwise non-privileged observations (Pinto et al., 2017; Chen et al., 2020). These factors are often introduced jointly within a single system. We instead study them separately by varying the skill prior, temporal and spatial feedback representation, and privileged-to-vision curriculum.

3 Problem Formulation

Agentic robotics for manipulation. A robot manipulation task class can be defined as a tuple $\mathcal{T} = \langle \mathcal{L}, \mathcal{E}, \mathcal{R}, \mathcal{O}, \mathcal{X}, \mathcal{B}, \mathcal{J} \rangle$: a natural-language instruction $\mathcal{L}$, a fixed workcell $\mathcal{E}$ with a known robot and sensor configuration $\mathcal{R}$, a bounded set of objects $\mathcal{O}$ with known models, a state space $\mathcal{X}$ of robot configurations and object poses, a belief space $\mathcal{B}$ giving the distribution over initial object poses, and a reward $\mathcal{J}$, here the per-object success indicator $\mathbb{I}(\text{success})$. A task instance $\tau_i = \langle o_i, x_i \rangle$ is drawn from $\mathcal{O}$ and $\mathcal{B}$. The agent produces a policy π represented as executable source code composed from reusable robot skills. Following the graph-based policy representation of GaP (Chen et al., 2026), we view π as a computation graph whose nodes invoke skills and whose edges define their execution and control flow. The policy is constructed over a skill library $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_g \cup \mathcal{S}_\tau \cup \mathcal{S}_A$, where $\mathcal{S}_0$ contains basic robot and sensor primitives; $\mathcal{S}_g$ contains reusable general-purpose skills; $\mathcal{S}_\tau$ contains task-specific skills provided by a human expert; and $\mathcal{S}_A$ contains task-specific skills synthesized by the agent. Given $\mathcal{S}$, an LLM agent A constructs π by selecting, parameterizing, and composing available skills, while implementing new skills when the existing library is insufficient.

On a task instance τ_i, the resulting policy $\pi(a \mid \mathcal{I})$ maps observations $\mathcal{I}$ to robot actions a in closed loop, and the resulting episode is evaluated by $\mathcal{J}$. Given a task class $\mathcal{T}$ and skill library $\mathcal{S}$, the agent seeks to construct a policy that succeeds across the belief space: $\pi^* = \arg\max_\pi \ \mathbb{E}_{x_i \sim \mathcal{B}} \left[\mathcal{J}\big(\pi(a \mid \mathcal{I})\big) \right]$, using only $\mathcal{L}$, $\mathcal{S}$, and the outcomes of its own policies. Policy improvement may therefore modify both the composition of the policy graph and the implementation of its constituent skills.

Self-learning loop. Unlike reinforcement learning, feedback in the self-learning loop is not restricted to a scalar reward. At iteration k, the current policy π_k runs in simulation on the training instances $\{\tau_i\}$. A feedback function F turns these runs into a textual (and optionally visual) report $f_k = F(\pi_k, \{\tau_i\})$. The agent then edits its own policy, conditioned on the report and on its edit history $\pi_{k+1} = A\big(\pi_k,\, f_k,\, \{(\pi_j, f_j)\}_{j<k}\big)$. The loop runs for K iterations. Success is reported on a held-out set of instances for the initial policy π_0 (zero-shot, k=0) and for the iterate with the best validation score, π_{k^*} with $k^* = \arg\max_k \text{val}(\pi_k)$. Held-out success at every iteration gives a learning curve for each setting and task. Each study below varies one component of this loop: the agent A, the skills $\mathcal{S}$, the feedback F, or the curriculum over training instances.

Assumptions and Scope. We assume access to a reliable physics simulator. Training and evaluation remain within the same manipulation task class, with a fixed robot and sensor configuration and bounded scene variation. We evaluate generalization to held-out instances of these task classes and do not study sim-to-real transfer or generalization to new embodiments.

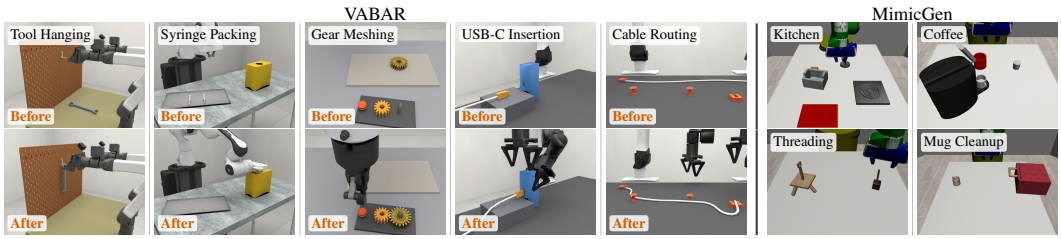

Figure 3: Five VABAR tasks (Left) and four MimicGen tasks (Right) used for this study.

4 Study Design

Task Setup. We evaluate five simulated manipulation tasks from the Variational Automation Benchmark for Agentic Robotics (VABAR) Xie et al. (2026b): (1) *Tool Hanging*, (2) *Syringe Packing*, (3) *Gear Meshing*, (4) *USB-C Insertion*, and (5) *Cable Routing*. For the curriculum study, we evaluate four multi-step tasks from MimicGen Mandlekar et al. (2023) using the Medium (D1) layouts: (1) *Kitchen*, (2) *Coffee*, (3) *Threading*, and (4) *Mug Cleanup* (Figure 3).

Skill Setup. We use open-robot-skills provided by Chen et al. (2026) as task-agnostic skill library $\mathcal{S}$. The library contains a set of 30 skills in the Agent Skills format, each with a natural-language `SKILL.md` describing when the skill applies and conditions. The library covers perception, grasping and motion planning primitives. The full list of skills can be found in Appendix B.

Metrics and Evaluation scale. In evaluation, every agent setting runs the same closed-loop episodes on the same training, validation and testing task instances with repeated runs. We report the success rate of the hold-out testing task instance that the scene configuration is different from the training and testing task instances. We run each of the ten agent settings on the five tasks with five independent runs per task with two rounds of feedback. In each round the current program runs on 4 training, 5 validation and 10 held-out instances. We use full spatial and temporal feedback and no curriculum for the skill composition study and spatial temporal skill expansion study. We use Opus 5 xhigh as default for the feedback and skill-library studies. Including pilot studies, we use 17,900 GPU-hours on NVIDIA RTX PRO 6000 Blackwell GPUs and 2,700 GPU-hours on NVIDIA A100 with in total 27,600 coding agent rollouts with an estimation of $70,000 computing cost. The study takes 2,500 coding agents editing sessions with $36,500 for LLM inference cost.

5 Experiments

5.1 Agent Model

We study how the choice of coding model affects both initial policy generation and subsequent self-learning. Each agent is given the shared general skill library, and the task-specific expert skills provided with VABAR(Xie et al. (2026b)). The skill library is sufficiently expressive for successful task execution. Each agent generates a zero-shot policy (@0) and then revises its code from execution feedback for two rounds (@1 and @2). We evaluate Claude Opus 5 (Anthropic, 2026b), Claude Opus 5.5 (Anthropic, 2026c), Claude Sonnet 5 (Anthropic, 2026d), Claude Fable 5.1 (Anthropic, 2026a), GPT-6 Astra (OpenAI, 2026a), GPT-6 Sol (OpenAI, 2026b), DeepSeek V4 (DeepSeek-AI, 2026), and Kimi K3 (Kimi Team, 2026). Table 1 reports held-out success after each round, and Figure 5 analyzes how models construct their initial policies from provided and generated skills.

Table 1 shows zero-shot performance is not sufficient to evaluate if the model can self-improve. Several models begin from nearly identical initial success rates but diverge after re-

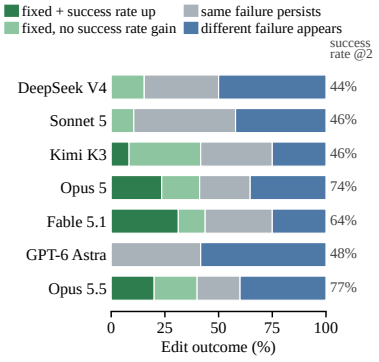

Figure 4: Outcomes of edits targeting a reported failing skill.

Model	Overall			Δ Overall		Tool Hanging			Syringe Packing			Gear Meshing			USB-C Insertion			Cable Routing		
	@0	@1	@2	1	2	@0	@1	@2	@0	@1	@2	@0	@1	@2	@0	@1	@2	@0	@1	@2
Human	82			–		87			90			100			87			47		
Opus 5, xhigh	45	64	74	+19	+10	22	62	60	28	38	35	100	100	100	0	50	100	73	72	73
Opus 5, high	41	62	**81**	**+21**	**+19**	25	**72**	**97**	40	30	45	67	100	100	0	33	87	75	76	76
Opus 5, low	57	67	74	+10	+7	0	30	57	**51**	50	**60**	100	100	100	60	80	80	73	76	76
DeepSeek V4	38	43	44	+5	+1	0	0	0	20	29	30	100	100	100	0	10	12	70	77	77
Sonnet 5	45	46	46	+1	0	0	0	0	32	34	34	100	100	100	25	25	25	67	72	72
Kimi K3	39	43	46	+4	+3	0	0	2	30	47	55	97	97	97	0	0	0	69	71	73
Fable 5.1	44	57	64	+13	+7	24	32	32	49	49	56	67	100	100	0	25	50	79	79	79
GPT-6 Astra	43	47	48	+4	+1	0	0	0	20	25	30	100	100	100	15	30	30	80	80	80
Opus 5.5	**67**	**74**	77	+7	+3	25	40	55	46	50	50	100	100	100	**85**	**100**	100	79	79	79
GPT-6 Sol	33	41	53	+8	+12	0	0	0	30	30	30	50	50	50	0	40	100	**83**	**83**	**83**

Table 1: Self-learning on VABAR by model. Held-out per-object success (%) reached within 0, 1, and 2 rounds of feedback (@k) reported by mean over the runs scored at each round. Δ_k denotes the gain in overall success from the k-th round of feedback, i.e. @k − @$(k-1)$.

ceiving the same execution feedback. For example, Opus 5 xhigh, Fable 5.1, GPT-6 Astra, and Sonnet 5 start at 43–45% overall success, yet after two rounds Opus 5 xhigh and Fable 5.1 improve to 74% and 64%, respectively, while GPT-6 Astra and Sonnet 5 remain at 48% and 46%. The same distinction appears across individual tasks: models that initially fail Tool Hanging or USB-C Insertion can sometimes repair these policies substantially, whereas others remain near their initial performance despite observing the same failures. Opus 5.5 is strong at both stages, increasing from 67% zero-shot to 77% after feedback. These results suggest that generating a plausible initial manipulation policy and using execution evidence to diagnose and repair that policy are distinct model capabilities (Figure 4).

Finding 1. Differences in model capability are less apparent at policy initialization but become pronounced during self-learning. Models with similar zero-shot performance can diverge substantially after receiving the same execution feedback, suggesting that policy generation and policy repair are distinct capabilities.

5.2 SKILL

We next study how agents use the skills they are given. The skill library provides an important prior: the expert skills encode task-specific manipulation strategies, while the general library provides reusable perception, grasping, and motion-planning routines. Such a library could help the agent by providing useful building blocks, or could instead constrain it to behaviors already represented in the library. To distinguish these cases, we trace the source of each line in the zero-shot policy (@0). After normalizing each line, we match it verbatim against the expert task-specific skills, the general-library skills also used by the human policy, and the remaining general-library skills; unmatched lines are attributed to new code written by the agent.

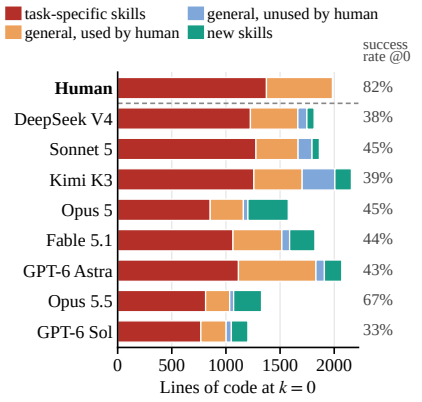

Figure 5: Zero-shot policy composition and attribution by code source.

Figure 5 reports the mean number of lines from each source, averaged over runs per task and then over tasks, and relates it to success after feedback in Table 1. As shown in Figure 5, models draw on the same skills in different ways. Models that improve most with feedback also write a larger share of their policy as new task-specific code: new code makes up 24% of the lines for Opus 5 xhigh, 18% for Opus 5.5, and 12% for Fable 5.1. They use the expert skills as a basis but add or modify task-specific logic where the provided implementations are insufficient. In contrast, Sonnet 5, DeepSeek V4, Kimi K3, and GPT-6 Astra write only 4–7% new code and rely more heavily on behavior already represented in the expert and general skills. These models also

Table 2: Effect of temporal feedback structure on held-out success (%). *Per motion* reports the outcome of each robot motion that is not aligned with the policy's semantic skills, *Per skill* reports the outcome of each semantic skills.

Feedback	Overall	Tool Hanging	Syringe Packing	Gear Meshing	USB-C Insertion	Cable Routing
Initial policy	3	0	0	0	0	17
Per episode	25	7	**90**	10	0	17
Per motion	36	45	55	**80**	0	0
Per skill	**52**	**90**	50	70	**20**	**27**

improve the least, ending at 39–46% overall success after two rounds. Using more of the library is not the same as adapting beyond it: Kimi K3 uses the largest share of general skills not used by the human policy (13% of its lines), yet still writes little new code and gains only 7 points. Since several of these models start from similar zero-shot success, the difference is not simply in the quality of the first policy, but in whether the agent can modify and extend the provided skills when the existing behavior is insufficient.

Finding 2. Expert skills are a boundary for weaker models but a scaffold for stronger ones. Weaker models largely compose behaviors already represented in the skill library, while stronger models selectively reuse and depart from human-provided skills to create new task-specific approaches.

5.3 Feedback

Temporal Failure Attribution. We next study whether self-learning agents can attribute an observed robot failure to the earlier policy decision that caused it. A manipulation episode consists of multiple perception, planning, and execution steps, so the point at which a failure becomes observable need not be the point at which the underlying error was introduced. As shown in Figure 6, the policy skill selected for repair can occur throughout the preceding trajectory, sometimes substantially before the point at which the episode finally halts. We therefore ask whether exposing intermediate execution outcomes helps the agent identify which earlier policy step should be revised.

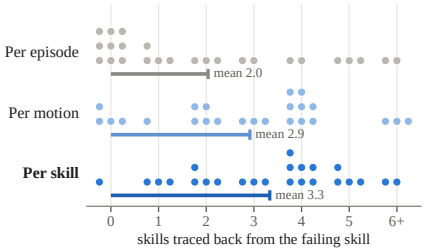

Figure 6: Number of skills traced back from the observed failing skill before each post-failure policy edit. Finer-grained and semantic feedback leads agents to trace failures further upstream.

Table 2 shows that the localized temporal feedback depends on where the underlying error occurs relative to where the failure becomes observable. Reporting every motion primitive can expose an earlier error once it manifests as incorrect motion, reaching 45% on Tool Hanging and 80% on Gear Meshing. However, this finer-grained feedback does not expose non-motion operations such as perception. The useful unit of feedback is therefore the intermediate outcome associated with a meaningful policy decision, rather than the finest available time step. In Tool Hanging, the per-episode agent repeatedly attributed failure to the final reaching stage and modified that part of the policy, reaching only 7% held-out success. The observed miss, however, originated from an earlier incorrect estimate of the wrench ring. Skill-level feedback records the state changes and return values around each semantic step, exposing this earlier error and increasing held-out success to 90%. Gear Meshing shows the same pattern, improving from 10% to 70%. Conversely, localized temporal feedback is less useful when the cause of failure is already apparent in the final state. In Syringe Packing, a syringe left outside or across the container opening directly exposes the relevant geometric failure, and per-episode feedback reaches 90% held-out success.

Finding 3. Feedback can be more useful when reported at the level of semantically meaningful policy skills, not simply at the finest temporal resolution. In multi-stage manipulation, the observed failure can occur well after the decision that caused it. Skill-level feedback exposes the intermediate outcomes of meaningful policy decisions and improves failure attribution, while finer motion-level feedback misses non-motion decisions such as perception.

Table 3: Effect of spatial representation on held-out success (%). World-frame and object-frame reports describe the same underlying scene state at different levels of relational organization, while part annotations add task-relevant functional geometry.

Spatial representation	Overall	Tool Hanging	Syringe Packing	Gear Meshing	USB-C Insertion	Cable Routing
Outcome only	19	15	45	0	0	33
State + relations, world frame	30	25	**60**	50	0	13
State + relations, object frames	40	63	50	60	0	**27**
+ part annotations	**52**	**90**	50	**70**	**20**	**27**

Spatial and Geometric Representation We next study whether the representation of execution state affects an agent's ability to reason about spatial failures. A conventional world-frame representation describes where objects are, but leaves the agent to reconstruct the object-relative quantities needed for manipulation. As suggests in Table 3, the world-frame state is sufficient in principle to recover the manipulation-relevant quantities, the agent often fails to reliably compose the required coordinate transformations and connect them to a policy decision. Object-frame feedback makes these same relationships explicit, such as grasp offsets and relative orientations, and substantially improves performance on Tool Hanging and, to a lesser extent, Gear Meshing. This suggests that information completeness alone is insufficient, and agents can possess the necessary coordinates yet still fail to derive the spatial relationships needed for manipulation. The same limitation extends beyond coordinate frames to object geometry itself. Providing accurate object state and geometry does not mean the agent understands how that geometry constrains action, because inferring the affordance features of the wrench and hanger from whole-object poses is challenging for agents. Part annotations further improve feedback by exposing the functional structures that matter for interaction. Part-annotated feedback reaches 52% overall success, compared with 40% for object frames and 30% for world-frame state and relations.



Figure 7: Spatial representations of the feedback report on one failed. The same execution state expressed at different levels of spatial organization.

Finding 4. Recovering task-relevant spatial and geometric relationships from raw state information remains a bottleneck for agentic robotics. Although world-frame and object-frame feedback describe the same underlying scene, agents perform substantially better when object-relative geometry and functional spatial relationships are explicitly available, rather than requiring them to derive them.

Feedback Reasoning. We next study how self-learning agents translate spatial understanding into executable robot motion. We compare how human-written and agent-learnt policies structure perception, motion planning, and recovery. Figure 8 shows that human policies frequently make geometric structure explicit through repeated perception and re-registration, intermediate waypoints, and dedicated motion-planning stages. These intermediate poses encode geometric priors such as approaching from a safer direction, maintaining clearance, or moving through configurations from which a feasible motion is easier to find. In contrast, self-learning agents more often use verification, retry, and recovery around coarser manipulation steps. They can reuse motions containing useful geometric structure, but are less consistent at reconstructing why that structure is necessary when generating or modifying the policy themselves.

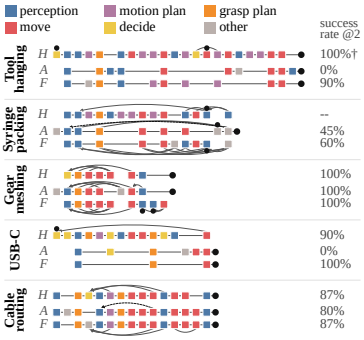

Figure 8: Policy structure by human(H), Astra(A), and Fable(F).

Syringe Packing provides a concrete example. The robot must carry a 150 mm syringe through a 55 mm opening while reasoning about the syringe as part of the moving system, surrounding obstacles, reorientation clearance, and insertion orientation (Fig-

ure 11). The human policy represents these dependencies explicitly with a collision world, held-object registration, collision-aware waypoints, and orientation-constrained insertion. In contrast, none of the agent runs without expert skills independently constructed an equivalent collision representation, and 17 of 19 initial self-written policies carried or released the syringe flat over the opening. Feedback often allowed agents to recognize the resulting errors and eventually adopt heuristics such as reorienting above the lid and descending vertically, but they still tended to encode these fixes as clearance constants, staging heights, and retries rather than as a general collision-aware model.

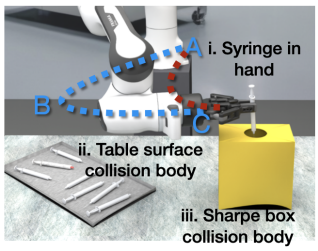

Figure 9: Human motion plan (blue) considers a collision free spot (B) to reorientate gripper, while agent plan (red) has limited room to reorientate.

Finding 5. Reasoning over collision geometry and spatial relationships remains a key bottleneck for agentic robot motion planning. Agents can recognize local geometric errors, but struggle to integrate object pose, held-object geometry, obstacles, and clearance into a coherent collision-aware motion plan.

5.4 Curriculum Design

We study a representative *privileged-to-perception curriculum*, a common design in reinforcement learning and used in prior works such as Pinto et al. (2017); Chen et al. (2020). As a focused case study of curriculum transfer in agentic robotics, the agent first learns with privileged simulation states with exact simulator geometry and is then transferred to non-privileged simulation states such as RGB-D perception. We evaluate this curriculum on four MimicGen tasks using their Medium (D1) layouts with 8 privileged and 8 perception iterations and Claude Fable 5.1. We also use three VABAR tasks: Tool Hanging (Level 2), Gear Meshing (Level 2), and USB-C Insertion (Level 1) with 5 privileged iterations and 5 perception iterations with Claude Opus 5. Figure 10 reports results on four MimicGen and three VABAR tasks. Exact task configurations, training and held-out layouts, iteration budgets, probe settings, and other protocol details are provided in Appendix J.

	# Skills		Success (%)	
Task	w/ curr.	w/o curr.	w/ curr.	w/o curr.
Tool Hanging	19.3	28.0	16	**63**
Gear Meshing	18.3	21.7	33	**40**
USB-C Insertion	18.0	10.7	**31**	16
Kitchen	13.0	13.3	33	**62**
Coffee	5.0	11.7	**76**	64
Threading	8.7	18.3	**86**	2
Mug Cleanup	6.3	9.3	**93**	77
Overall	12.7	16.1	**52**	46

Table 4: Policy size (# Skills, final policy) and held-out success (mean over the last three iterations) with (w/ curr.) and without (w/o curr.) the privileged-to-perception curriculum; mean over three runs.

Figure 10 suggests that privileged-to-perception learning does not uniformly improve over learning directly from perception. Privileged information removes perception uncertainty during early learning, but, as suggested by Findings 4 and 5, accurate geometry alone does not guarantee that the agent discovers the right spatial representation or manipulation strategy. The privileged phase can learn useful task structure, as in Threading, Coffee, and Mug Cleanup, but it can also sometimes commit early to brittle or non-generalizable choices. For example, Kitchen develops a grasp strategy that depends on geometric precision unavailable after transfer, while Tool Hanging remains below 10% held-out success during privileged learning despite access to exact wrench geometry. Thus, privileged information can simplify the learning problem without necessarily producing a policy that remains appropriate under perception. We further visualize and analyze representative failure cases in Appendix J.2.

In addition, Table 4 further suggests that once a policy structure is established under privileged state, subsequent perception-based learning tends to preserve and patch that structure rather than fundamentally reorganize it (which corresponds to Finding 1). Across tasks, the privileged-to-perception policies change little in size after transfer, from 12.0 skills at the end of the privileged phase to 12.7 after perception learning on average, while the final perception-only policies contain 16.1 skills. The resulting path dependence can be beneficial when the early structure captures a manipulation strategy that remains valid under vision, but this can be harmful when the structure assumes geometric

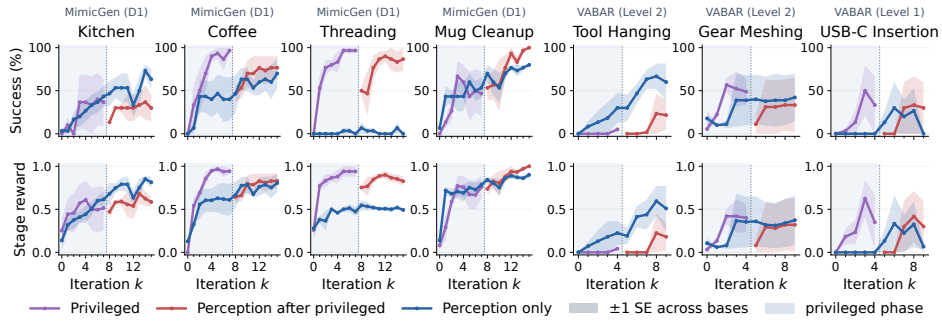

Figure 10: Held-out success (top) and stage reward (bottom) for the privileged-to-perception curriculum and perception-only control on MimicGen (left) and VABAR tasks (right), evaluated on 10 fixed held-out layouts per task. Lines show means over three seed replicates and bands show ± 1 standard error. The dotted line marks transfer to perception; shading indicates the privileged phase, during which only the curriculum experiment has access to privileged geometry.

precision unavailable from perception. In such cases, the curriculum does not merely provide an initialization; it can constrain subsequent self-learning.

Finding 6. Privileged-to-perception curriculum learning can either accelerate or constrain subsequent perception-based learning. Privileged agents learn policies that rely on exact state, usually with less verification and recovery, and often retain this structure after transfer. This helps when the learned manipulation strategy remains valid, allowing perception learning to focus on perception errors, but hurts when the policy depends on state precision unavailable under perception or lacks mechanisms for handling perceptual uncertainty.

6 Discussion and Limitations

Our results return to the central question of this study: what determines whether an agent can interpret physical interaction and convert it into effective robot-policy improvement? Figure 11 provides a complementary view across the experiments. Only 7.3% of the human-solvable Level-1 test cases are solved by zero-shot policy generation alone, but the results suggest that improving agentic robotics requires the combination of iterative self-learning with richer simulation feedback, human task skills, and more powerful models. At the same time, this decomposition shows that 4.1% of test cases solved by a human-written policy under the same robot and simulation setup remain unsolved by the agents. Moreover, this study primarily considers Level-1 VABAR tasks with uncluttered environments and relatively simple motions. The remaining gap is concentrated in agents' ability to interpret physical interaction into task-relevant reasoning and policy changes, particularly for spatial reasoning, manipulation planning, and collision-aware motion planning.

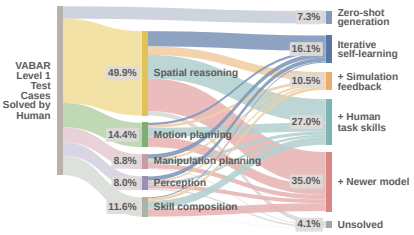

Figure 11: **What closes the gap to human-written policies?** For VABAR Level-1 test cases solved by the human policy with Opus 5, improvements come progressively with the earlier findings, while a small fraction unsolved.

Limitations. This study does not cover the full range of conditions that may affect agentic robot self-learning, such as prompting strategies and harness configurations. Given the large number of agent rollouts required for controlled evaluation, we leave a broader study of these factors to future work. Our model comparison is also necessarily a snapshot in time, as new agent models are released frequently; we plan to update the evaluation as new models become available. We focus on simulation to enable repeatable evaluation, leaving sim-to-real transfer and adaptation to real-world feedback as important next steps. Finally, we do not consider human-in-the-loop self-learning, where human feedback may complement autonomous simulation feedback during policy improvement.

AI USE STATEMENT

Generative AI tools were used to assist with implementing experiments and organizing experimental findings, and polishing portions of the manuscript. All AI-assisted analyses, text, and suggestions were reviewed and verified by the authors. Separately, because studying AI agents is central to the paper, we also use AI-produced outcome as ingredients for evaluation results. The LLM agents evaluated in this work are experimental subjects and are described as part of the methodology. The authors take full responsibility for the content, claims, and conclusions of this work.

REPRODUCIBILITY STATEMENT

Our experiments are built on publicly available simulation environments, open-source robot skill libraries, and commercially available coding agents. We specify the task setup, skill configurations, feedback design, training protocol, and evaluation procedure in the main paper and appendix. To support reproduction of our results, we will further release the self-learning harness, experiment configurations, evaluation scripts, and task-specific integration code used in this study upon publication.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pp. 287–318. PMLR, 2023.

Anthropic. Claude Fable 5.1 and Claude Mythos 5.1 System Card. https://www.anthropic.com/claude-fable-5-1-mythos-5-1-system-card, September 2026a. Accessed September 2026.

Anthropic. Introducing claude opus 5. https://www.anthropic.com/news/claude-opus-5, July 2026b. Accessed September 2026.

Anthropic. Claude Opus 5.5 System Card. https://www.anthropic.com/claude-opus-5-5-system-card, September 2026c. Accessed September 2026.

Anthropic. Claude Sonnet 5 System Card. https://www.anthropic.com/claude-sonnet-5-system-card, June 2026d. Accessed September 2026.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009. doi: 10.1145/1553374.1553380.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on robot learning*, pp. 66–75. PMLR, 2020.

Kaiyuan Chen, Shuangyu Xie, Letian Fu, Justin Yu, William Pacini, Sandeep Bajamahal, Hudson Kim, Jaimyn Drake, Daehwa Kim, Haoru Xue, Jonathan Francis, Christian Juette, Peter Schaldenbrand, Muhammet Yunus Seker, Ruwan Wickramarachchi, Uksang Yoo, Guanzhi Wang, Adithyavairavan Murali, Balakumar Sundaralingam, S. Shankar Sastry, Spencer Huang, Yuke Zhu, Linxi Fan, and Ken Goldberg. GaP: A graph-as-policy multi-agent self-learning harness for variational automation tasks. *Conference on Robot Learning (CoRL)*, 2026.

DeepSeek-AI. DeepSeek-V4 Preview: Entering the Era of Affordable Million-Token Context. https://deepseek.com/en/news/v4-preview/, April 2026. Accessed September 2026.

Letian Fu, Justin Yu, Karim El-Refai, Ethan Kou, Haoru Xue, Huang Huang, Wenli Xiao, Guanzhi Wang, Fei-Fei Li, Guanya Shi, Jiajun Wu, Shankar Sastry, Yuke Zhu, Ken Goldberg, and Linxi Fan. CaP-X: A framework for benchmarking and improving coding agents for robot manipulation. In *Proceedings of the International Conference on Machine Learning*, 2026.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pp. 32808–32824, 2024a.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, pp. 540–562. PMLR, 2023a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, pp. 1769–1782. PMLR, 2023b.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024b.

Kimi Team. Kimi K3: Open Frontier Intelligence. *arXiv preprint arXiv:2607.24653*, July 2026. URL https://arxiv.org/abs/2607.24653.

Chengshu Li, Cem Gokmen, Gilbert Levine, Roberto Martín-Martín, Sanjana Srivastava, Chen Wang, Joey Wong, Ruohan Zhang, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, 2022.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation*, pp. 9493–9500, 2023a.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International conference on robotics and automation (ICRA)*, pp. 9493–9500. IEEE, 2023b.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Runyu Lu, Yubo Wu, Ethan Kou, Letian Fu, Wenli Xiao, Ajay Mandlekar, Yinzhen Xu, Guanya Shi, Ken Goldberg, Ang Chen, Mosharaf Chowdhury, Yuke Zhu, Linxi "Jim" Fan, and Guanzhi Wang. ASPIRE: Agentic /skills discovery for robotics, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *Conference on Robot Learning (CoRL)*, 2023.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking LLMs for embodied decision making. In *NeurIPS 2024 Track on Datasets and Benchmarks*, 2024.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Michael Murray, Abhishek Gupta, and Maya Cakmak. Teaching robots with show and tell: Using foundation models to synthesize robot policies from language and visual demonstration. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4033–4050. PMLR, 2025.

OpenAI. GPT-6 Astra: A New Generation of Intelligence. https://openai.com/index/gpt-6-astra/, 2026a. Accessed 2026.

OpenAI. GPT-6 Sol. https://developers.openai.com/api/docs/models/gpt-6-sol, 2026b. OpenAI API model documentation, accessed September 2026.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *Robot: Science and Systemsn*, 2017.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. EMBODIEDBENCH: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.

William Shen, Nishanth Kumar, Sahit Chintalapudi, Jie Wang, Christopher Watson, Edward S. Hu, Jing Cao, Dinesh Jayaraman, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. TiPToP: A modular open-vocabulary planning system for robotic manipulation. *arXiv preprint arXiv:2603.09971*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

Wenli Xiao, Jia Xie, Tonghe Zhang, Haotian Lin, Letian "Max" Fu, Haoru Xue, Jalen Lu, Yi Yang, Cunxi Dai, Zi Wang, Jimmy Wu, Guanzhi Wang, S. Shankar Sastry, Ken Goldberg, Linxi "Jim" Fan, Yuke Zhu, and Guanya Shi. ENPIRE: Agentic robot policy self-improvement in the real world, 2026.

Shuangyu Xie, Kaiyuan Chen, Ziyang Chen, Simeon Adebola, Yixuan Huang, Zehan Ma, Tianshuang Qiu, Wentao Yuan, Dhruv Shah, Pannag R. Sanketi, and Ken Goldberg. Robovista: Evaluating vision-language models for diverse robot applications. In *Robotics: Science and Systems (RSS)*, 2026a.

Shuangyu Xie, Kaiyuan Chen, Uksang Yoo, Xinyue Zhu, Meadow Shen, Jaimyn Drake, Eugen Solowjow, Lionel Gulich, Alexander Millane, Rafael Cathomen, Tuur Stuyck, Sangeeta Subramanian, Yichao Pan, Bruce McGowan, Michael Benedict, Linxi Fan, Yuke Zhu, Spencer Huang, and Ken Goldberg. VABAR: A benchmark for robots that repeat one job across a known distribution of variation. https://vabar-benchmark.github.io/, 2026b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023.

Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Wang. VLMbench: A compositional benchmark for vision-and-language manipulation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.