SKILL-FORMING EVIDENCE FOR LLM AGENTS: JOINT SELECTION AND TARGET-MODEL-AWARE COMPILATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agents increasingly turn stored trajectories into reusable skills. Recent work has begun to show that trajectory composition can materially change downstream skill quality, alongside continued progress in skill extraction. We ask the next structural question: what makes a set of trajectories sufficient for forming a transferable skill, and how should that evidence be represented for the target model? We formulate skill induction as two coupled decisions over experience: *selection*, which chooses the trajectories that jointly identify transferable structure, and *compilation*, which expresses that selected evidence in a form the target model can use. In an exact Conditional Procedure World, six machine-verified propositions show that skill-forming value can be irreducibly joint, that sparse support sets can preserve full-pool utility, and that the preferred evidence can change with the target model. Neural experiments reproduce these mechanisms across Pythia, Qwen2.5, and SmolLM2. Closed-loop tests then show that these data decisions matter for agent performance. On ALFWorld, changing the selected trajectories moves success from 0.164 to 0.351, and the gain from success-only selection replicates across five prospective experience pools. Raw delivery of the same failure-inclusive evidence reduces success by 0.157 relative to compilation. On TextWorld, a compiler that encodes episode-varying recipe content hurts transfer, whereas an invariant-aware compiler reverses the effect. Together, these results characterize skill-forming evidence as a jointly structured and target-model-dependent object: effective skills require informative trajectory sets and a representation that preserves their transferable structure for the model that uses the skill.

1 INTRODUCTION

LLM agents increasingly convert stored trajectories into reusable skills: executable code libraries (Wang et al., 2024a), workflow memories and programmatic skills (Wang et al., 2025b;a; Zheng et al., 2025), and trace-consolidation or skill-optimization systems (Ni et al., 2026; Yang et al., 2026b; Guo et al., 2026). Much of this progress has improved the *extraction operator*: how trajectories are summarized, merged, verified, distilled, or optimized into a reusable artifact. Recent work has also begun to study the experience pool itself. SkillLens shows that success/failure composition changes downstream skill utility under a fixed extractor and that the same skill can yield different gains across target models (Huang et al., 2026). SkillGen uses contrast between successful and failed trajectories during synthesis (Ma et al., 2026), while Trace2Skill aggregates trajectory-local lessons from broad execution pools (Ni et al., 2026). These results establish experience composition as an important part of the skill lifecycle.

The broader history of LLM development makes this progression familiar. Data pruning, high-quality alignment data, and curated "textbook" corpora can change model behavior while the model family remains fixed (Sorscher et al., 2022; Zhou et al., 2023; Gunasekar et al., 2023). For skill induction, the analogous data object is a finite pool of trajectories. The central question is therefore structural: *what makes a set of trajectories sufficient for forming a transferable skill?*

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.

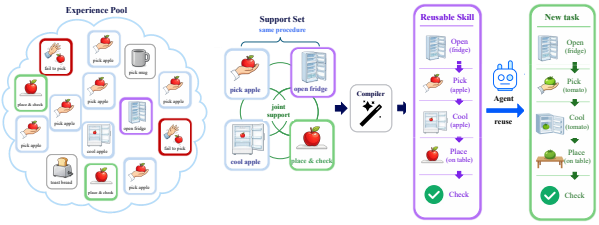


Figure 1: Overview of the two-stage skill-induction problem. A support set retains complementary experiences whose joint evidence identifies transferable structure. A compiler then expresses that structure in a form the target model can execute. The experiments follow this sequence from exact identification to neural use and closed-loop transfer.

A single trajectory rarely specifies an entire reusable procedure. Several records may need to be recognized as evidence about the same procedure. Their differences then constrain what that procedure does across contexts. Two individually ordinary trajectories can therefore become decisive together. A large collection of near-duplicates can add little once the same constraints are already represented. We call the subset that preserves the procedure-defining information the *skill-forming evidence set*. Its value is a property of the set and the relations among its members.

A second issue appears when this evidence reaches the target model. A trajectory is both evidence about a latent procedure and visible behavior inside the model's context. These roles can produce different effects. A failed action can exclude a wrong hypothesis, yet the displayed action can also attract probability mass during in-context use. Skill induction therefore contains two coupled decisions. *Selection* chooses the trajectories that jointly identify transferable structure. *Compilation* expresses that structure in a form the target model can reliably use. Figure 1 summarizes this formulation.

The paper follows this logic through three questions. **First, what structure makes a trajectory set skill-forming?** We construct an exact Conditional Procedure World (CPW) in which every trajectory subset can be evaluated. This exposes higher-order evidence interactions, alignment requirements, and minimal support sets. **Second, how does the target model realize the value of that evidence?** We transfer the same structural manipulations to Pythia, Qwen2.5, and SmolLM2. We then vary raw and compiled interfaces as model competence changes. **Third, do these choices change closed-loop agent performance?** ALFWorld varies selected trajectories under a fixed compiler and varies representation at fixed selection. TextWorld changes the cross-episode invariant and tests which structure the compiler must preserve.

Contributions. We make three connected contributions. (1) We formalize *skill-forming evidence* as a set-level object. Exact analysis shows irreducibly joint value, a requirement for cross-record alignment, and sparse support sets that preserve the functional content of much larger pools. (2) We show that evidence value depends on the target model and its interface. The same records can support exact identification yet produce poor raw in-context behavior, and this relationship changes with model competence. (3) We isolate the closed-loop effects of selection and compilation. ALFWorld shows that trajectory composition changes agent success under a fixed compiler. TextWorld shows that compilation transfers when it preserves structure that remains stable across target episodes. Together, these results characterize the structure that makes experience reusable as a skill and the interface required to realize that structure in a target model.

2 Related Work

Trajectory-to-skill extraction and lifecycle studies. Agent systems transform experience into reusable artifacts through code libraries, workflow memories, programmatic skills, trace consolidation, and skill optimization (Wang et al., 2024a; 2025b;a; Zheng et al., 2025; Ni et al., 2026; Yang et al., 2026b; Guo et al., 2026). SkillGen uses successful and failed trajectories to induce and verify candidate skills (Ma et al., 2026). Trace2Skill extracts trajectory-local lessons and consolidates

them into transferable guides (Ni et al., 2026). Skill lifecycle studies broaden the analysis to retrieval, content quality, target models, and experience composition (Yang et al., 2026a; Liu et al., 2026a;b; Huang et al., 2026). SkillLens is especially close to our motivation: it fixes an extractor while varying success/failure composition and also measures how the same skill transfers across target models (Huang et al., 2026). Those experiments establish that composition and target-model effects are real. Our analysis characterizes the structure behind these effects through exact set-level interactions, minimal support sets, and matched selection–interface interventions.

Data selection and sufficient evidence. Data-centric language-model results show that selection and curation can materially alter learning at fixed model families (Sorscher et al., 2022; Zhou et al., 2023; Gunasekar et al., 2023). Demonstration selection, coresets, and data valuation emphasize similarity, diversity, coverage, or per-example influence (Liu et al., 2022; Xin et al., 2026; Sener & Savarese, 2018; Ghorbani & Zou, 2019). UCS further estimates subset-level unseen coverage for in-context demonstration selection (Xin et al., 2026). Machine teaching asks for examples sufficient to identify a target concept (Goldman & Kearns, 1995; Zhu et al., 2018). Our support-set formulation brings this identification view to trajectory sets. Complementary records can acquire value through interaction. The selection target is therefore a jointly sufficient evidence set whose value can be distributed across records.

Target-model semantics and compilation. In-context learning depends on demonstration structure, binding, prompt format, label priors, and copying mechanisms (Brown et al., 2020; Olsson et al., 2022; Lu et al., 2022; Zhao et al., 2021). Agent-memory studies report domain-dependent differences between raw and consolidated experience (Shen et al., 2026; Zhang et al., 2026; Hu et al., 2026), while SkVM compiles already-formed skills for heterogeneous model–harness capability profiles (Chen et al., 2026). We study the selection–reader interface directly. Proposition 6 makes the preferred evidence set depend on the reader, the neural experiments measure selection–interface interactions across model competence, and TextWorld identifies cross-episode invariance as the property successful compilation must preserve. Appendix D gives the expanded comparison.

3 METHOD

3.1 SKILL-FORMING EVIDENCE AND TWO-STAGE INDUCTION

Let a trajectory pool T contain attempts at procedures that may be reused on future tasks. Skill induction maps this pool to an artifact that guides future behavior. We separate the mapping into two decisions. *Selection* chooses a subset $S \subseteq T$ whose joint evidence preserves the transferable distinctions needed by the procedure. *Compilation* maps S to an artifact K_S that exposes those distinctions in a form the target model can interpret and execute.

The two decisions share the same evidence. Selection determines which constraints on the latent procedure are retained. Compilation determines how those constraints are expressed. The value of a selected record is realized when its role in the evidence set survives this interface. This motivates two measurements: the interaction structure of trajectory sets and the size of a subset that preserves the induced skill.

3.2 CONDITIONAL PROCEDURE WORLD

The Conditional Procedure World (CPW) makes the evidence structure of skill induction exactly observable. An agent sees trajectories generated by an underlying conditional procedure and must recover a rule that transfers to evaluation contexts. The construction separates three ingredients that are entangled in real logs: surface alignment across records, context–action–outcome constraints within records, and the target reader that converts selected evidence into behavior.

A CPW instance is $W = (\mathcal{P}, \mathcal{C}, \mathcal{A}, R^*, p_{\text{eval}})$. A procedure $\rho \in \mathcal{P}$ is rendered as $L{=}10$ surface tokens; each token is independently paraphrased with probability ν into one of $V{=}6$ variants, and distinct procedures share no tokens. Two renderings are alignable when their Jaccard token overlap is at least $\theta{=}0.35$. Contexts are bit-vectors $\mathcal{C} = \{0,1\}^n$ ($n{=}2$; $n{=}3$ in the shortcut world), actions are $\mathcal{A} = \{0,1,2\}$, and $R^* : \mathcal{C} \to \mathcal{A}$ is the ground-truth rule. A trajectory $\tau = (\rho, \sigma, c, a, y)$ records one attempt with rendering σ, context c, action a, and outcome $y = \mathbf{1}[a = R^*(c)]$.

The reference extractor E clusters trajectories by alignability, retrieves the largest cluster aligned with the evaluation procedure, and forms the version space $\mathcal{V}(S)$ (Mitchell, 1982). Success rows impose $t[c] = a$ and failure rows exclude $t[c] = a$. The extractor returns the unique minimal-complexity table when one exists, marks conflicts when $\mathcal{V}(S) = \emptyset$, and otherwise uses a majority-success fallback. Writing $K_S = E(S)$ for the extracted artifact, exact utility is

$$U(S) = \sum_c p_{\text{eval}}(c) \Pr[K_S(c) = R^*(c)], \tag{1}$$

with uniform fallback worth $1/3$.

3.3 Joint Evidence and Skill Support Sets

Because $U(S)$ is available for every subset, the Harsanyi dividend and order spectrum are exact:

$$\Delta(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} U(B), \qquad D_k = \sum_{|A|=k} \Delta(A). \tag{2}$$

We use $\sum_{k \geq 3} D_k$ as higher-order functional interaction mass. A skill support set is any smallest subset preserving full-pool utility within tolerance ϵ:

$$S^*(\epsilon) \in \arg\min_{S \subseteq T} \{|S| : U(S) \geq U(T) - \epsilon\}. \tag{3}$$

The interaction spectrum measures whether functional value is distributed across combinations of trajectories. The support set then asks which of those trajectories are necessary to preserve the induced skill.

3.4 Evaluation Ladder

We test the same mechanism at three levels. The exact world isolates the structure of trajectory sets and provides complete subset-level measurements. Neural experiments present matched evidence structures to Pythia, Qwen2.5, and SmolLM2. They test whether the exact-world relations survive language-model representation and whether their realized value changes with the target model. Closed-loop experiments use agent-generated trajectories and evaluate the resulting skill by interactive task success. Evidence-budget controls separate structure from volume. Fixed-selection comparisons isolate compilation. Fixed-compiler comparisons isolate selection. Prospective pools test whether selection effects persist across newly collected experience.

4 Results

The results follow the three questions introduced above. Section 4.1 identifies the set structure that carries a reusable procedure. Section 4.2 tests how a target model realizes that evidence. Section 4.3 then measures the consequences of selection and compilation in interactive agents.

4.1 Joint and Sparse Skill-Forming Evidence

We begin with the structure of the evidence set. Pool size and success rate are held fixed while relations among records are changed. Propositions 1–5 characterize how local observations become a jointly identifiable procedure.

Proposition 1 (Identification; minimal evidence) *$U(S)$ depends on S only through its cluster partition and evidence rows. Minimal identifying success-row counts are 1 (constant rule), 2 (single-bit rule), and 4 (XOR rule); no 3-row success set identifies XOR.*

Proposition 2 (Pure higher-order interaction exists) *For the XOR success cover $\{\tau_1, \ldots, \tau_4\}$, $U(\emptyset) = \frac{1}{3}$, U of every singleton, pair, and triple is $\frac{1}{2}$, and $U(\text{full}) = 1$. The exact dividend spectrum is $D_1 = \frac{2}{3}$, $D_2 = -1$, $D_3 = \frac{2}{3}$, $D_4 = \frac{1}{3}$: all positive mass beyond first order sits at orders 3–4. Moreover certain pairs are confidently identified as a wrong single-bit rule.*

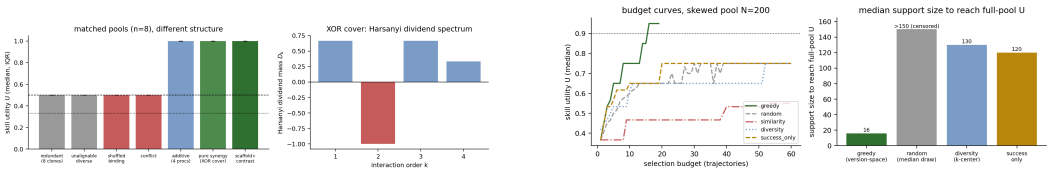


Figure 2: Exact-world structure of skill-forming evidence. **Left:** matched $n{=}8$ pools show that skill utility changes with relations among records; the XOR spectrum places positive interaction mass at orders 3–4. **Right:** on the skewed $N{=}200$ pool, version-space greedy preserves full-pool utility with 16 trajectories, whereas similarity-based selection leaves a utility gap at the same budget. Horizontal reference lines mark the best-singleton ceiling (.5) and the uniform fallback ($1/3$) on the left, and full-pool utility on the right.

The XOR construction gives the simplest example of set-level data value: no proper subset changes utility above $1/2$, yet the completing record raises the full set to 1. The negative D_2 strengthens with the learner's simplicity preference, moving from $D_2 \geq 0$ under weak preference to $-0.20 \to -1.0$ under decisive Occam selection, while order-≥ 3 interaction mass remains positive under continuous utilities. Partial data can therefore support a simple but incomplete rule until complementary evidence completes the structure (Blumer et al., 1987).

Proposition 3 (Scaffold necessity: impossibility) *Let the observation channel destroy cross-trajectory alignment and bit attribution (private tokens; per-trajectory context relabeling). Then there exist worlds w_1 (rule on bit 1) and w_2 (rule on bit 2) with identical pool laws, hence for every extractor, $\min_w U_w(\text{pool}) \leq \frac{3}{4} < 1$, while with intact scaffold both worlds are identified exactly ($U = 1$).*

Proposition 4 (Greedy support-set approximability) *Hypothesis-elimination coverage $g(S)$ is integer-valued, monotone, and submodular with $g(\emptyset){=}0$; greedy row selection reaches full coverage $g_{\max}{:=}g(\text{pool})$, hence the full-pool version space and, when that version space uniquely identifies the extracted rule, the same skill and utility, within a factor $1 + \ln g_{\max}$ of the optimal cover by the submodular-cover bound (Wolsey, 1982). (U itself is a step function; the approximation guarantee applies to the coverage surrogate.)*

Proposition 5 (Redundancy–Scaffold Law) *If no two trajectories of S are alignable, then*

$$U(S) \leq \max_{\tau \in S} U(\{\tau\}). \tag{4}$$

With a shared scaffold, joint-evidence pools can exceed this singleton ceiling (1 versus $\frac{1}{2}$ in Proposition 2).

Together, these propositions separate two roles of data composition. Shared structure makes records attributable to the same latent procedure; complementary variation supplies the constraints that distinguish that procedure from alternatives. The pre-registered $n{=}8$ battery changes these relations while holding pool size and success rate fixed. Scaffolded-contrast and XOR-cover pools reach $U{=}1.0$, redundant and unalignable pools remain at the singleton ceiling 0.5, additive pools have zero cross-procedure dividend mass, and conflicts are detected in 30/30 runs.

The same logic predicts that a large pool can contain a much smaller skill-forming core. On $N{=}200$ trajectories over five rule families, version-space greedy reaches full-pool utility with a median of approximately 16 trajectories. As N grows $50 \to 100 \to 200 \to 400 \to 800$, median support size changes $7 \to 13 \to 16 \to 19 \to 19$ while the support ratio falls from 0.14 to 0.024. Across eight rule families, support size tracks the minimal identification count with Spearman $\rho{=}1.0$. An $81\times$ larger hypothesis class raises the budget by $1.5\times$ while still reaching the full-pool skill; a class excluding the true rule saturates at $U{=}.50$. Table 1 reports the matched selector comparison.

These results identify a skill-forming set by the constraints it contributes. A trajectory gains value when it completes information missing from the current set. Once the required constraints are represented, the pool can grow without increasing the functional content of the induced skill. Appendix A gives the scaffold, contrast, boundary, and shortcut tests that isolate this mechanism.

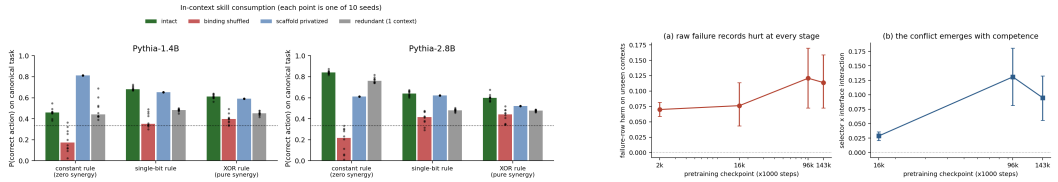


Figure 3: Set-level evidence structure persists under neural readers, while its realized value changes with model competence. **Left:** the exact-world manipulations carry over to Pythia: binding shuffles collapse performance, complementary coverage beats redundancy, and scaffold privatization weakens conditional-rule use; dotted lines mark chance ($1/3$). **Right:** across Pythia-2.8B pretraining, raw-failure harm and the selector–interface interaction change with model competence.

4.2 Target-Model Dependence of Evidence Value

The first result identifies which trajectory sets contain the transferable rule. The selected evidence is then read by a language model. Its realized value can change at this interface. Proposition 6 gives an exact case in which two readers prefer different subsets of the same pool.

Proposition 6 (Target-model-conditioned selection) *Let reader A interpret failure records as exclusion constraints and reader B interpret every displayed action as endorsed. There is a pool and budget at which the U_A- and U_B-optimal subsets are disjoint and* $\max_S \min(U_A(S), U_B(S)) \leq \min(\max_S U_A, \max_S U_B) - \frac{1}{4}$.

Exhaustive evaluation of the 924 budget-6 subsets of a 12-record biased pool gives $\max U_A{=}1.0$ for subsets retaining all four failure rows, $\max U_B{=}.75$ for success-only subsets, and a joint ceiling of .50. The subset that most sharply identifies a rule can therefore differ from the subset that a raw display-copying target uses most effectively.

Real language models reproduce the structural part of the exact result. Pythia-1.4B and 2.8B receive verbalized trajectory records in context. Shuffling recorded actions reduces consumption in all 30/30 seed–rule comparisons per scale (within-rule sign tests, $p{=}.002$), full-cover pools beat budget-matched redundancy in all 20/20 conditional-rule comparisons per scale (within-rule $p{=}.002$), and privatizing renderings reduces conditional-rule performance in 19/20 comparisons at 1.4B and 17/20 at 2.8B. At a fixed budget, a four-record support set beats four-record redundant subsets in all 20/20 conditional-rule comparisons per scale (within-rule $p{=}.002$). A deliberate non-cover preserves the unseen context for the inferable single-bit rule and leaves XOR ambiguous.

We then vary selection and representation factorially. On one noisy mixed pool with budget 6, compilation moves elimination-greedy selection from .397 to .440 and positive coverage from .618 to .542, producing a median per-instance interaction of $+0.121$ at 1.4B. The 2.8B replication gives $+0.099$, and a budget sweep at 4/6/8/10 keeps the interaction positive ($+.07/ + .07/ + .08/ + .14$). The interaction links the two stages directly. Selection determines which constraints enter the candidate skill. Compilation determines how the target model receives those constraints. Table 4 reports the factorial cells.

Failure trajectories expose this coupling most clearly. In a biased-context experiment, success-only evidence gives .34 accuracy on unseen contexts. Adding failure trajectories raises exact-world identifiability but lowers Pythia consumption to .22 at 1.4B, with the same harmful direction at 2.8B. On unseen contexts, the model assigns .80 probability mass to actions displayed inside the failure records. Compiling the negative evidence into positive constraints restores failure-context accuracy. The same informational content therefore produces different behavior when its interpretation changes.

That interpretation also changes with model competence. Qwen2.5-1.5B-Instruct and SmolLM2-1.7B-Instruct reproduce failure harm and positive selector–interface interactions under plain and chat renderings. At Qwen2.5-3B the failure effect approaches zero; at 7B adding clean failure records improves performance by .13 in plain rendering and .15 in a token-matched comparison ($p \leq .016$), indicating a shift from display copying toward outcome-label use. In the label-sensitive regime, mislabeled high-elimination rows become especially damaging: positive coverage reaches

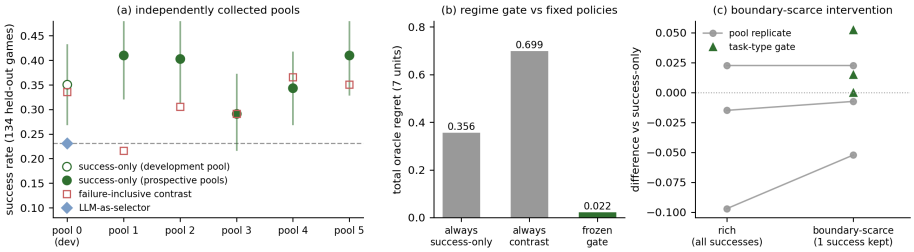


Figure 4: Closed-loop effects across experience pools and data regimes. **(a)** The development pool and five prospective ALFWorld pools preserve the positive success-only-versus-no-skill direction (prospective-pool median $+.172$); the dashed line marks the no-skill baseline ($.231$). **(b–c)** Pre-specified secondary analyses show how selector and compiler choices vary with the evidence regime; details are reported in Appendix C.

1.00 versus $.43$ for raw elimination and $.50$ for compiled elimination ($+.57$, d_z=3.4, p=.002). Evidence value is therefore conditioned by both the structure of the selected set and the target model that reads it.

Audited SkillsBench logs (Li et al., 2026) show that these evidence regimes also occur in realistic experience pools: $.44$ of tasks have zero successes even after pooling nine runs, and same-model failures are more similar than cross-model failures on 92% of 61 eligible tasks ($p < 10^{-4}$). Appendix B reports the complete real-log, multi-step, and corruption analyses.

4.3 Closed-Loop Effects of Selection and Compilation

The exact and neural results identify the structure of skill-forming evidence and its dependence on the target model. We next test whether these effects change interactive task success. ALFWorld separates selection from representation. TextWorld then tests which information compilation must preserve across episodes.

4.3.1 ALFWorld: Selection Under a Fixed Compiler

A Qwen2.5-7B-Instruct agent collects 160 ALFWorld train-split episodes across six task types with 23% success (Shridhar et al., 2021). For each task type, one of four blind selectors chooses k=4 trajectories (random, success-only, diversity, contrast), or uses the full 13–46-trajectory pool. One fixed deterministic compiler converts the selected evidence into a skill, and the same greedily decoded agent executes 134 paired OOD games.

Changing the selected data under the fixed compiler produces a 19-point spread in execution success. Success-only selection with compilation raises the no-skill baseline from $.231$ to $.351$ (24W/8L, p=.0077, 95% CI $[.037, .201]$). The full pool reaches $.291$, random $.209$, diversity $.164$, contrast $.336$, and an ExpeL-style self-consolidation baseline $.276$. Under the fixed compiler, skill quality varies strongly with the selected set and is non-monotonic in stored volume.

The entire chain was repeated on five newly collected experience pools under the pre-specified protocol. All five preserve the positive success-only-versus-no-skill direction: $+.179$, $+.172$, $+.060$, $+.112$, and $+.179$ (development pool $+.119$), with median $+.172$. Four of five paired bootstrap CIs exclude zero, and all 36 leave-one-task-type-out gaps remain positive. Figure 4 summarizes this pool-level replication; the secondary regime diagnostics are analyzed in Appendix C.

4.3.2 ALFWorld: Compilation at Fixed Selection

The same experiment also holds the selected trajectories fixed and varies how they are delivered. Delivering the stored failure-inclusive contrast selection raw lowers success by $.157$ relative to compiled delivery (32W/11L, p=.002, 95% CI $[.067, .246]$); success-only compilation adds $.082$ (p=.087). The selected set determines the available procedural evidence. Compilation determines whether that evidence becomes usable guidance during execution.

4.3.3 TEXTWORLD: COMPILATION AROUND THE TRANSFERABLE INVARIANT

ALFWorld establishes a closed-loop representation effect. TextWorld cooking then changes the transferable structure by varying the recipe across episodes (Côté et al., 2018). Under the same pre-specified protocol (three recipe variants, disjoint train/eval seeds, paired n=60), the standard compiler reaches at most .317 against a .400 no-skill baseline. Success-only compiled records trail the same raw records by $-.133$ (0W/8L, p=.008, 95% CI $[-.217, -.050]$). The compiler extracts a modal procedural skeleton that names ingredients appearing in only 1–4 of 20 evaluation games per type, so the artifact encodes an episode-varying dimension as transferable content.

A pre-specified follow-up changes the abstraction rule while keeping selection fixed. Removing ingredient and recipe names raises success-only compilation from .267 to .500 ($+.233$, p=.0015; $+.100$ over raw) and contrast from .267 to .600 ($+.333$, p=.00015; $+.250$ over raw, p=.00075), which is .200 above no-skill. Together, the two domains separate the roles of selection and compilation. Selection determines which experience-derived constraints are available. Compilation transfers them when it preserves structure that remains stable across the target episode distribution.

5 DISCUSSION

Recent lifecycle studies show that experience composition can change the utility of an extracted skill (Huang et al., 2026). Our results provide a set-level account of this dependence. Skill-forming information can be distributed across several trajectories. Shared structure links records to the same procedure. Complementary variation then removes alternative procedures that remain consistent with any one record. Evidence value therefore depends on which constraints become available together. This explains both the higher-order interactions in CPW and the sparse support sets recovered from larger pools. A growing pool can contain many additional trajectories while adding few new constraints to the procedure that must be identified.

This set-level view gives a concrete objective for experience selection. Useful selection should preserve the distinctions that determine future behavior. Similarity, diversity, success rate, and pool size can correlate with those distinctions, but their value depends on the structure already present in the selected set. The support-set results make this dependence explicit: a trajectory becomes important when it supplies a constraint that the current set is missing. Once that constraint is represented, another trajectory carrying the same information contributes less to identification. Selection can therefore be evaluated by the functional structure preserved by the set. Independent trajectory scores provide a less complete view when value appears only after several records are combined.

The selected set is then interpreted by a target model. This introduces a second source of variation. Our neural results show that informational value and realized value can diverge at this interface. Failure trajectories provide useful boundary information for exact identification, yet smaller language models may copy the displayed failed action. As model competence increases, outcome labels become more usable, and clean failures can become helpful. Corrupted high-constraint records then become more damaging because the model can use their labels more effectively. The selector–interface interaction follows from the same mechanism. Selection determines which constraints enter the candidate skill. Compilation determines the semantics through which the target model receives those constraints.

The closed-loop experiments connect both mechanisms to agent behavior. In ALFWorld, changing the selected trajectories under one fixed compiler substantially changes success, and the effect replicates across newly collected pools. Holding selection fixed then isolates the effect of representation. TextWorld identifies the transfer criterion for that representation. A reusable artifact should preserve structure that remains stable across the episodes in which the skill will be used. Encoding episode-varying recipe content reduces transfer. Changing the abstraction while keeping the selected trajectories fixed restores performance. The two domains therefore connect set-level evidence structure to the final execution of the skill: selection determines what transferable constraints are available, and compilation determines whether those constraints survive the interface and the change of episode.

The same decomposition also clarifies how skill-induction methods should be evaluated. A single end-to-end success score can combine two different failures: the selected pool may omit a necessary constraint, or the compiled artifact may present the right constraint in a form the target model cannot

use. Set-level coverage and closed-loop execution therefore answer different questions. Reporting both makes it possible to locate whether improvement comes from better evidence selection, better representation, or both. This separation becomes especially useful when comparing systems across model scales, because the evidence structure can remain fixed while the model's interpretation of that evidence changes.

These results suggest a practical design sequence for skill systems. First, define the transferable distinctions that future behavior must preserve. Second, select trajectories whose joint evidence identifies those distinctions. Third, compile that evidence around the invariants of the target episode distribution and the interpretation capabilities of the target model. This sequence also exposes clear scaling questions. Learned selectors can search large experience stores for jointly sufficient evidence. Learned compilers can adapt the representation to model competence and transfer context. Evaluation can then measure both the functional coverage of the selected set and the execution value of the compiled artifact. Richer hypothesis classes, uncertain task identity, and larger closed-loop domains are natural settings for testing how far this formulation extends.

6 Conclusion

Skill-forming evidence is a structured set-level object. Exact analysis shows that its value can be irreducibly joint and sparsely supported. Neural experiments show that the realized value of the same evidence depends on the target model and its interface. Closed-loop experiments show that both trajectory selection and invariant-aware compilation change agent success. These findings motivate a two-stage view of skill induction: select the experiences that jointly identify transferable structure, then compile that structure into a representation the target model can reliably execute.

The broader significance is a change in how the skill-induction pipeline is designed. The experience pool, the extraction process, and the target-model interface form three linked parts of the same system. This view gives skill systems a principled data objective: preserve the constraints that define reusable behavior, then express them at the level of invariance required for transfer. As agents accumulate larger stores of experience, this distinction becomes increasingly important. Larger stores create more opportunities for reuse and also increase redundancy, conflicting evidence, and model-dependent interpretations. Identifying the small amount of jointly sufficient structure inside those stores can make skill induction more data-efficient and more reliable. It also suggests a route toward data scaling laws for agent skills: measure how the amount and structure of sufficient evidence change as task complexity, experience volume, and target-model competence grow. These quantities provide a common basis for comparing future selectors and compilers by the transferable behavior they preserve.

AI use statement

Generative AI tools were used to assist with literature search, provide feedback on experimental design, support result interpretation, assist with code implementation and debugging, and support manuscript drafting, translation, revision, and grammar editing. The research hypotheses and core conceptual framing were developed and refined by the authors. All AI-assisted outputs incorporated into the work were reviewed and verified by the authors, who take responsibility for the final content.

Reproducibility statement

The environments, selectors, extractors, statistical analyses, and data-figure generators are released as code; experiment scripts freeze their declared predictions and thresholds in module docstrings before the run (the pre-registration document indexes them), write raw per-row outputs alongside aggregated statistics, and use fixed seeds. Real-model results use publicly available checkpoints (Pythia, Qwen2.5, SmolLM2), and the real-log studies use the audited public SkillsBench leaderboard trajectories (Li et al., 2026).

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *International Conference*

on Learning Representations, 2023.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, James L. McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Le Chen, Erhu Feng, Yubin Xia, and Haibo Chen. Skillrt: Compiling skills for efficient execution everywhere. *arXiv preprint arXiv:2604.03088*, 2026.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. TextWorld: A learning environment for text-based games. *CoRR*, abs/1806.11532, 2018.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, 2019.

Sally A. Goldman and Michael J. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, 1999.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Zhongxin Guo, Danrui Qi, Hanwen Gu, Peng Cheng, and Yongqiang Xiong. Skill-disco: Distilling and compiling agent traces into reusable procedural skills. *arXiv preprint arXiv:2606.26669*, 2026.

Qisheng Hu, Quanyu Long, and Wenya Wang. When continual learning moves to memory: A study of experience reuse in LLM agents. *arXiv preprint arXiv:2604.27003*, 2026.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Zisu Huang, Jingwen Xu, Yifan Yang, Ziyang Gong, Qihao Yang, Muzhao Tian, Xiaohua Wang, Changze Lv, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Xue Yang, Dongdong Chen, Xiaoqing Zheng, and Chong Luo. From raw experience to skill consumption: A systematic study of model-generated agent skills. *arXiv preprint arXiv:2605.23899*, 2026.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, Chujun Tao, Binxu Li, Xuandong Zhao, Hejia Geng, Xiaojun Wu, Junwei Zhou, Xiaokun Chen, Hanwen Xing, Yubo Li, Qunhong Zeng, Di Wang, Yuanli Wang, Roey Ben Chaim, Penghao Jiang, Haotian Shen, Luyang Kong, Xinyi Liu, Runhui Wang, Xuanqing Liu, Jiachen Li, Xin Lan, Yueqian Lin, Wengao Ye, Junwei He, Songlin Li, Yue Zhang, Yipeng Gao, Yijiang Li, Ze Ma, Liqiang Jing, Tianyu Wang, Kaixin Li, Yiqi Xue, Haoran Lyu, Yizhuo He, Yuchen Tian, Shutong Wu, Bowei Wang, Yixuan Gao, Bo Chen, Litong Liu, Sikai Cheng, Jiajun Bao, Shuaicheng Tong, Shuwen Xu, Terry Yue Zhuo, Tinghan Ye, Qi Qi, Miao Li, Longtai Liao, Zelin Tan, Chang Shi, Xilin Tang, Srinath Tankasala, Boqin Yuan, Yaoyao Qian, Jianhong Tu, Chenguang Wang, Yizhou Sun, Wei Wang, Aaron Taylor, Ziyue Yang, Changkun Guan, Zhikang Dong, Xinyu Zhang, Steven Dillmann, Han-chung Lee, and Dawn Song. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026.

Hongyi Liu, Haoyan Yang, Tao Jiang, Bo Tang, Feiyu Xiong, and Zhiyu Li. Skillsvote: Lifecycle governance of agent skills from collection, recommendation to evolution. *arXiv preprint arXiv:2605.18401*, 2026a.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022.

Yujian Liu, Jiabao Ji, Li An, Tommi Jaakkola, Yang Zhang, and Shiyu Chang. How well do agentic skills work in the wild: Benchmarking LLM skill usage in realistic settings. *arXiv preprint arXiv:2604.04323*, 2026b.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Yuchen Ma, Yue Huang, Han Bao, Haomin Zhuang, Swadheen Shukla, Michel Galley, Xiangliang Zhang, and Stefan Feuerriegel. Skillgen: Verified inference-time agent skill synthesis. *arXiv preprint arXiv:2605.10999*, 2026.

Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Lloyd S. Shapley. A value for n-person games. In *Contributions to the Theory of Games II*, pp. 307–318. Princeton University Press, 1953.

Maohao Shen, Kaiwen Zha, Zexue He, Zhang-Wei Hong, Siru Ouyang, J. Jon Ryu, Prasanna Sattigeri, Suhas Diggavi, and Gregory Wornell. Decocted experience improves test-time inference in LLM agents. *arXiv preprint arXiv:2604.04373*, 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.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners. In *International Conference on Learning Representations*, 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*, 2024a.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024b.

Zora Zhiruo Wang, Apurva Gandhi, Graham Neubig, and Daniel Fried. Inducing programmatic skills for agentic tasks. In *Conference on Language Modeling*, 2025a.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *International Conference on Machine Learning*, 2025b.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, 2015.

Paul L. Williams and Randall D. Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022.

Jiayi Xin, Xiang Li, Evan Qiang, Weiqing He, Tianqi Shang, Weijie J. Su, and Qi Long. UCS: Estimating unseen coverage for improved in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026.

Min Yang, Jinghua Piao, Xu Xia, Xiaochong Lan, Jiaju Chen, Yongshun Gong, and Yong Li. Skillmaster: Toward autonomous skill mastery in LLM agents. *arXiv preprint arXiv:2605.08693*, 2026a.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026b.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Dylan Zhang, Yanshan Lin, Zhengkun Wu, Yihang Sun, Bingxuan Li, Dianqi Li, and Hao Peng. Useful memories become faulty when continuously updated by LLMs. *arXiv preprint arXiv:2605.12978*, 2026.

Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 2021.

Boyuan Zheng, Michael Y. Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, and Yu Su. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *International Conference on Learning Representations*, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N. Rafferty. An overview of machine teaching. *arXiv preprint arXiv:1801.05927*, 2018.

A Exact-World Evidence Structure

The main text uses CPW to establish the first claim: the data that forms a skill is jointly structured and can be sparsely supported. This appendix provides tests that isolate the two ingredients behind that claim—cross-record alignment and complementary constraints—and the selector comparisons used to quantify support size.

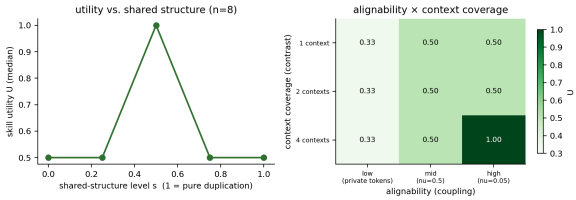


Figure 5: Alignment and complementary variation jointly determine exact-world skill formation. **Left:** utility follows an inverted-U pattern as shared structure changes at fixed budget; the endpoints correspond to pure diversity (s=0) and pure duplication (s=1), and the peak is the scaffold-plus-contrast pool. **Right:** the high-coupling, full-contrast condition reaches U=1.0, while weaker alignment or contrast leaves the procedure underidentified.

Proof sketches and exact verification. All six propositions are finite consequences of the CPW definitions. For Proposition 1, the extractor and K_S are determined by the cluster partition and the context–action–outcome evidence rows; exact version-space enumeration gives identifying success counts 1, 2, and 4 for the constant, single-bit, and XOR rules. Proposition 2 follows by evaluating all 2^4 subsets of the four-row XOR cover and applying Equation 2. For Proposition 3, private renderings and per-record context relabeling make the bit-1 and bit-2 worlds induce the same observed-pool law. The two rules agree on two of four contexts and disagree on two, so any common prediction has average utility at most $3/4$ across the two worlds and therefore $\min_w U_w(\text{pool}) \leq 3/4$. For Proposition 4, let $\mathcal{H}$ be the finite hypothesis class and $C_\tau \subseteq \mathcal{H}$ the hypotheses excluded by row τ. Then $g(S) = |\bigcup_{\tau \in S} C_\tau|$ is an integer-valued monotone submodular coverage function; matching full-pool coverage matches its version space, after which the Wolsey submodular-cover bound gives the stated approximation factor. Proposition 5 follows because pairwise unalignability leaves every aligned cluster a singleton, so the extractor cannot combine constraints across records and is bounded by the best singleton. Proposition 6 follows by exact evaluation of all $\binom{12}{6} = 924$ budget-6 subsets: the reader-specific optima are disjoint, their maxima are 1.0 and .75, and the best joint value is .50, yielding the stated .25 gap.

A.1 Alignment, Scaffold, and Higher-Order Value

XOR is a canonical synergistic construction in information theory. Our measured quantity is utility-level Harsanyi interaction. The redundancy relevant to Propositions 3 and 5 is alignment redundancy: repeated observable structure that makes separate records attributable to one underlying procedure. Shared structure supplies identity; contrastive variation supplies constraints on the procedure.

The XOR-cover pool is compared with a matched pool whose per-trajectory evidence is unchanged but whose cross-trajectory scaffold is destroyed through private tokens and context relabeling by hidden XOR masks. The intact condition has higher-order dividend mass 1.0. In the knockout, the full pool never exceeds its best singleton over 30/30 seeds and total order-≥ 2 mass is ≤ 0; either knockout component alone breaks identification in at least 97% of seeds. The manipulation isolates attribution across records as the mechanism that makes joint evidence usable.

At fixed budget n=8, pure duplication gives U=0.5, pure diversity gives U=0.5, and scaffold-plus-contrast gives U=1.0 (margin 0.5). A 3×3 grid independently varies alignability and context coverage. Utility peaks in the high-alignability, high-contrast cell, showing that repeated structure and complementary variation make separate contributions to the same identifying set.

A.2 Support-Set Baselines

The main-text scaling result asks whether the functional content of a growing pool can be preserved by a smaller subset. Table 1 compares version-space selection with coverage-, success-, similarity-, diversity-, facility-location-, and random baselines on identical pools and the same utility target.

selector	median support size	note
version-space greedy	**16 / 200**	ratio 0.08
coverage-only greedy	30	censored 17%
success-only	120	
facility location	130	censored 37%
diversity (k-center)	130	
random (median draw)	> 150	censored 48%
max-similarity	—	gap 0.48 at budget 16

Table 1: Support size required to reach full-pool utility ($U{=}0.90$), using identical pools and evaluator across selectors (median over 30 seeds). Version-space greedy reaches the target with 16 of 200 trajectories; the remaining selectors require larger budgets or retain a utility gap at the same budget.

The support-size scaling result is therefore aligned with the exact identification result: the selector retains rows that remove remaining hypotheses, while data-volume and similarity criteria can spend budget on constraints already represented elsewhere in the pool.

A.3 Boundary, Binding, and Shortcut Tests

With demonstrations biased to half the context space, success-only identification chooses an incorrect rule on 50/50 seeds ($U{=}0.5$); adding naturally occurring failure trajectories restores $U{=}1.0$. The failure rows supply boundary constraints absent from the successful demonstrations. A binding control then permutes outcome labels while preserving text, contexts, actions, and token counts; correct identification falls from 50/50 to 0/50, locating the useful signal in the context–action–outcome relation.

A 3-bit shortcut world compares identification with frequency summarization. When a spurious bit is perfectly correlated with the causal bit, the frequency extractor follows the shortcut and reaches $U{=}0.0$ under covariate flip. The identification extractor remains underdetermined, and two decorrelating support rows restore $U{=}1.0$. On the matched $n{=}8$ battery, the frequency extractor remains at 0.5 on the two pools whose positive utility appears at interaction orders ≥ 3.

B Target-Model and Interface Extensions

The main text establishes the second claim by showing that selected evidence can change value across target models and interfaces. This appendix expands that claim in five steps: coverage-controlled selection, selector–interface interactions across model competence, extraction-to-execution transfer, realistic evidence regimes, and robustness to corrupted evidence.

B.1 Coverage-Controlled Support Selection

A skewed 16-record pool with context counts $7/7/1/1$ makes complete coverage by a uniform random four-subset rare (probability .05; observed 0% over 40 draws). Aggregate support-versus-random margins are $-.002$ (5/10 seeds) for the bit rule and $+.028$ (8/10) for XOR. Per-context analysis explains the small aggregate differences. An uncovered context remains inferable for the bit rule (median .61) and becomes ambiguous for XOR (median .30). The support set's own weakest context has median .24 with one record per context, motivating coverage budgets defined at the context level.

B.2 Selector–Interface Coupling and Target-Model Competence

The factorial values underlying the main-text interaction are reproduced in Table 4. The checkpoint and instruction-tuned replications then place the interaction on a competence axis.

	rule	full8	support4	random4	redund.4
1.4B	bit	.681	.611	.619	.467
	XOR	.610	.545	.515	.438
2.8B	bit	.640	**.647**	.623	.490
	XOR	.600	**.572**	.586	.476

Table 2: Median $P(\text{correct action})$ under fixed context budgets (10 seeds). Support sets outperform redundant subsets in all 20/20 seed–rule comparisons per scale. At 2.8B, support4 is within .028 of full8 for both rules; at 1.4B, a larger gap remains.

	rule	supp.4	nocov.4	nocov: covered	nocov: missing
1.4B	bit	.593	.641	.512	**.820**
	XOR	.555	.572	.638	**.248**
2.8B	bit	.733	.739	.517	**.862**
	XOR	.647	.703	.865	**.152**

Table 3: Coverage double dissociation at budget 4 (per-context medians). The unseen context remains inferable for the single-bit rule and is three-way ambiguous for XOR. Across both model scales, probability mass follows whether the missing context is structurally inferable; the dissociation grows from .57 to .71.

Under Qwen2.5-1.5B-Instruct and SmolLM2-1.7B-Instruct, failure records reduce unseen-context accuracy by paired means $+.16/+.13$ in plain rendering and $+.14/+.07$ in chat (10/10 seeds in every cell, p=.002, $d_z = 1.3$–2.2). The raw-interface penalty of elimination selection is $+.19/+.26$ in plain rendering and $+.32/+.15$ in chat ($d_z = 2.0$–5.4); selector $\times$ interface interactions range from $+.06$ to $+.18$ (all $p \leq .035$). Qwen2.5-3B reduces the failure effect to $+.10$ in plain rendering (p=.21) and zero in chat. At 7B, adding clean failure records improves performance by .13 in plain rendering and .15 in a token-matched comparison ($p \leq .016$), while mislabeled high-elimination rows yield .43 raw and .50 compiled against 1.00 for positive coverage ($+.57$, $d_z = 3.4$, p=.002).

B.3 Extraction-to-Execution Transfer

The model is also evaluated as its own extractor. Stage A reads the pool and stores per-context argmax actions as a compiled skill; Stage B receives that skill without the source trajectories, using $0.31\times$ the tokens. Intact pools compile with fidelity $1.0/.75$ for the bit rule and $.75/.50$ for XOR at 1.4B/2.8B, and support-4 pools compile identically to their full pools. Redundant pools compile to fidelity .33 on unseen contexts. Native record-format rendering raises following of the same compiled table to 1.00, giving end-to-end utility $1.00/.75$ for bit/XOR. A cross-model 2×2 comparison shows that downstream use tracks table fidelity across compiler identities. In these transfer checks, 2.8B extraction fidelity is lower than 1.4B.

A three-stage workflow (inspect $\rightarrow$ conditional middle tool $\rightarrow$ submit) tests whether the evidence mechanism extends beyond one-step action prediction. Under an abstract stage-name query, completion is .00 at both Pythia scales. Under native trace completion, the same evidence yields .50–.75 median completion, with a paired interface effect of $+.60$ per rule (p=.002, $d_z > 3.4$). Compiling model executions into four workflow lines preserves completion at $.53\times$ the source tokens. At 2.8B, a non-covering pool yields 1.0 completion on the inferable bit-rule context and .50 on XOR. Qwen2.5-1.5B-Instruct reaches native support-4 completion $1.0/.75$ for bit/XOR, support exceeds redundancy, a binding shuffle moves bit completion from 1.0 to .00, and the missing-context dissociation is 1.0 for bit versus .00 for XOR.

B.4 Realistic Evidence Regimes

On 23 mixed-outcome SkillsBench tasks (Li et al., 2026), five selectors choose k=4 records for a held-out run pair. Qwen2.5-1.5B-Instruct predicts verifier outcomes with chance .5. Success-only

selector	raw records	compiled
version-space greedy	.397	.440
positive coverage	**.618**	.542
random	.463	.451
similarity	.485	.484

Table 4: Selector $\times$ interface factorial for one pool family (budget 6, Pythia-1.4B, median $P(\text{correct})$, 10 instances). The $+0.121$ interaction, the median of per-instance interactions, shows that the effect of compilation depends on which evidence the selector retains.

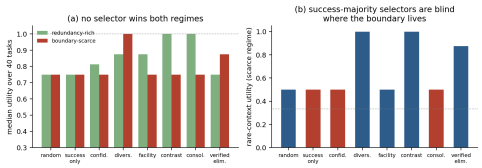


Figure 6: Selection depends on the evidence regime in 40 prospective task instances (budget 8 from 100 corrupted logged trajectories; task = unit). **(a)** Success-majority consolidation reaches median utility 1.0 when every context contains success mass and .75 when rare contexts are represented by honest failures; the regime $\times$ selector interaction is $+.21$, CI $[.09, .33]$; the dotted line marks the full-pool median. **(b)** When rare contexts are carried by failures, failure-inclusive and coverage-spreading selectors recover boundary information that success-majority selection misses; the dotted line marks chance $(1/3)$.

selection gives $P(\text{correct}) = .18$ on held-out failures versus .81 on success probes; failure-inclusive selection reaches .56, a median per-task gap of $+.37$ on 23/23 tasks ($p < 5\times10^{-5}$). The model-scale comparison changes the direction of the failure-probe effect: the gap is $-.14$ at 3B ($p = .03$) and $+.60$ at 7B on 23/23 tasks. This pattern extends the main-text observation that the usable meaning of failure evidence changes with the target model.

A second pre-registered probe predicts the next tool call along held-out successful runs. Raw records reduce next-tool accuracy relative to no evidence for every selector (1.5B $-.07$ to $-.09$, $p \leq .008$; 3B $-.10$ to $-.13$, $p \leq .001$; 7B $-.06$ to $-.11$). Compiling the same evidence into a 26-token procedural plan (approximately 205 tokens in raw form) changes the result by $+.03/+.11$ at 1.5B/3B ($p \leq .03$). The execution probe therefore extends the interface effect from action prediction to multi-step tool use.

A prospective blind-selection study uses 40 task instances from eight rule families, each with 100 logged trajectories containing 10% verifier false positives and a 5% correlated duplicated wrong record. Selection uses budget 8 and a fixed noise-tolerant compiler, with task as the statistical unit. When every context contains success mass, success-majority consolidation reaches median utility 1.0 and consensus-verified elimination trails random in the registered comparison. When rare contexts are carried by honest failure rows, success-majority selection gives rare-context utility .50, while diversity reaches 1.0 and verified elimination .875. The regime $\times$ selector interaction is $+.21$ with bootstrap CI $[.09, .33]$ and task-level sign-flip p=.002.

B.5 Robustness to Corrupted Evidence

The corruption sweep separates verifiable and self-consistent errors. Consensus-gated elimination maintains full utility through the tested 10–30% range of one-sided label noise, whereas correlated duplication defeats consensus and bootstrap-stability checks from 10%. Observation corruption is likewise unresolved from the stored pool because the corrupted observation and its label remain mutually consistent. These results define the evidence-generation regimes in which high-constraint selection remains reliable.

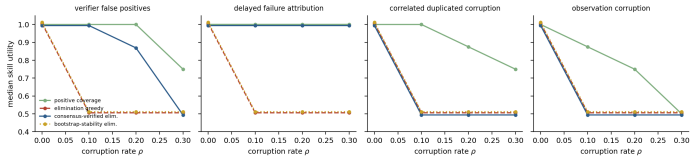


Figure 7: Selection reliability under non-iid corruption (median utility, 20 seeds per cell). Consensus-verified elimination preserves full utility across the tested one-sided label-noise range. Correlated self-consistent duplication reduces the value of within-pool verification from ρ=.1, while positive coverage degrades more gradually under correlated and observation corruption.

Selection	Raw	Compiled	Δ interface
Success-only	.269	.351	+.082
Contrast	.179	.336	+.157
Δ selection	+.090	+.015	

Table 5: Selection-by-interface factorial on the same 134 held-out ALFWorld games (no-evidence baseline .231). Cell values report the registered arms, and the margins summarize the selection and interface differences. Paired interface tests are reported in §4.3.2 (p=.087 and p=.002).

C Closed-Loop Robustness and Statistical Record

The main text uses ALFWorld and TextWorld to establish the third claim: selection and compilation both change closed-loop behavior. This appendix reports the matched factorial, related ALFWorld comparisons, boundary analyses, and the inferential status of the results.

C.1 ALFWorld Selection–Interface Factorial

The four ALFWorld arms sharing the same 134 paired episodes form a descriptive selection-by-interface 2×2. Compilation changes success by $+.082$ under success-only selection and $+.157$ under contrast; at the raw interface, success-only exceeds contrast by $+.090$, while the compiled difference is $+.015$. The cell values and paired tests are reported in Section 4.3.2.

C.2 Matched ALFWorld Comparisons

The remaining ALFWorld arms vary subset size, selection criterion, semantic selection, consolidation, representation, and pool identity around the same headline comparison. Table 6 collects these comparisons using the values already reported in the corresponding experiments.

C.3 Boundary Analysis of Selector and Compiler Policies

A pre-specified follow-up maps how compilation policy changes with properties of the training pool. A variance-abstracting compiler reaches .343 on ALFWorld against .231 no-skill but reaches .317 on TextWorld against .400. A second rule uses the instruction-grounding rate g, the fraction of action arguments appearing in the episode goal: $g \geq .30$ selects success-only with the standard compiler, and $g < .30$ selects contrast with the abstracting compiler. Across seven evaluation units its oracle regret is .022, compared with .356 for always-success-only and .699 for always-contrast; four of 384 fixed assignments match or improve on that regret. In these seven units, g is confounded with domain identity, so the result is treated as exploratory.

A within-ALFWorld scarce-pool intervention reduces three of six task types to one success plus all failures. Contrast exceeds success-only in one of three pool replicates, below the registered two-of-three crossover criterion. The finer interaction is $+.017$, and a task-type coverage gate is best in two of three replicates. These outcomes define the observed boundary of the regime analysis.

Factor	Control	Result	Where
Subset sufficiency	random k=4	.209, below no-skill .231	§4.3.1
Coverage/diversity	diversity k=4	.164, below no-skill	§4.3.1
Experience volume	full pool	.291, below k=4 at .351	§4.3.1
Failure evidence	contrast vs. success-only	.336 vs. .351; regime-dependent	§4.3.1, §C.3
Semantic selection	LLM-as-selector	.231, at no-skill	Table 7
Pool consolidation	self-consolidation (ExpeL-style)	.276, below .351	§4.3.1
Selection/interface	raw vs. compiled interface	compilation adds +.082/+.157	§4.3.2
Abstraction choice	mismatched vs. invariant-aware	interface effect negative → positive under both selections	§4.3.3
Automatic compiler	automatic variance compiler	helps ALFWorld, fails TextWorld	§C.3
Pool replication	five prospective pools	5/5 positive, median +.172	§4.3.1
Task-type robustness	task-type leave-one-out	36/36 gaps positive	§4.3.1
Context-length control	consumer-tokenizer audit	winner is shortest compiled arm	this table

Table 6: Matched ALFWorld comparisons around the main condition (success-only selection, compiled, .351 vs. no-skill .231). The rows summarize how subset size, selection criterion, representation, pool identity, and task family relate to the same closed-loop result.

Result	Status	Unit (N)	Effect	Inference
CPW propositions + battery	exact, machine-verified	pools/seeds	P1–P6	24/27 registered checks pass
Target-model interpretation	confirmatory, pre-specified thresholds	paired seeds	copy → label trust	Holm/BH within family
ALFWorld skill vs. no-skill	development + pre-specified extension	game (134)	.231 → .351 [.037, .201]	p=.0077, survives Holm
Untouched 74-game tranche	prospective directional	game (74)	+.095	p=.117
Five experience pools	prospective replication	pool (5)	5/5 positive, median +.172	consistency; one-sided sign p=.031
Fresh selector seeds	robustness	skill (4)	+.037 to +.134	3 of 4 at $p\leq$.007
Contrast compiled vs. raw	confirmatory	game (134)	+.157 [.067, .246]	p=.002, survives Holm
TextWorld compiled below raw	confirmatory (failure branch)	game (60)	−.133 [−.217, −.050]	p=.008, survives Holm
Invariant-aware rescue	pre-registered intervention	game (60)	+.250 over raw	p=.00075, survives Holm
LLM-as-selector	pre-specified baseline	game (134)	−.119 vs. blind success-only	p=.001
Self-consolidation	exploratory baseline	game (134)	.276 vs. .351	ns
Automatic variance compiler	pre-specified follow-up, mixed	game	ALFWorld .343; TextWorld .317	domain-dependent, descriptive
Regime gate	exploratory prospective probe	unit (7, 2 domains)	regret .022 vs. .356/.699	rank 4/384, descriptive
Pool-level crossover	registered boundary test	pool (3)	crossover in 1 of 3	criterion unmet
Finer regime analyses	registered secondary	pool (3)	+.017; gate best 2 of 3	directional only

Table 7: Statistical record for the main and boundary results, including evaluation unit, effect size, preregistration status, and inferential treatment.

C.4 Multiplicity and Evidence Status

Inferential p-values are grouped by the pre-registered families used in the real-log selection probes, real-log execution probes, ALFWorld closed loop, and TextWorld compilation chain. Holm family-wise and Benjamini–Hochberg corrections include all pre-specified inferential tests in each family. All 13 results stated as significant remain below $\alpha = .05$ under both corrections; the largest Holm-adjusted value is $p = .041$. Pooling the 13 tests into one conservative family yields the same maximum adjusted value. Directional, robustness, and consistency analyses retain their declared inferential status.

D Expanded Related Work

D.1 Skill Extraction and Lifecycle Studies

Agents retain experience as executable skills (Wang et al., 2024a), distilled cross-task insights (Zhao et al., 2024), trajectory exemplars (Zheng et al., 2024), verbal reflection (Shinn et al., 2023), retrieval-organized memory (Packer et al., 2023; Park et al., 2023), and bootstrapped traces (Zelikman et al., 2022; Huang et al., 2023); see Wang et al. (2024b). Recent systems synthesize reusable artifacts through workflow memory (Wang et al., 2025b), programmatic skill construction (Wang et al., 2025a; Zheng et al., 2025), trace consolidation (Ni et al., 2026), control-flow compilation (Guo et al., 2026), and optimizer-style skill refinement (Yang et al., 2026b). SkVM compiles already-formed skill artifacts for heterogeneous model–harness targets from capability profiles (Chen et al., 2026). Our compilation stage operates on selected trajectory evidence, linking target-reader semantics to the cross-episode invariants retained in the induced skill.

Several recent papers also study the experience supplied to these extractors. SkillGen uses success/failure contrast to identify reusable patterns and failure modes before verifying the induced skill (Ma et al., 2026). Trace2Skill extracts local lessons from a broad pool and hierarchically consolidates them (Ni et al., 2026). Skill lifecycle methods manage creation, refinement, selec-

tion, and governance (Yang et al., 2026a; Liu et al., 2026a). Skills-in-the-Wild and SkillLens study downstream utility at larger scale (Liu et al., 2026b; Huang et al., 2026). SkillLens directly varies success/failure composition under a fixed extractor and evaluates identical skills across different target models. It therefore establishes two empirical facts that motivate our analysis: pool composition affects skill utility, and skill utility depends on the target model. Our work characterizes the set structure that can produce these effects. The exact world exposes higher-order interactions and minimal support sets, while the neural and closed-loop experiments isolate how selected evidence interacts with its representation.

Recent comparisons of raw and consolidated memory provide a related view of representation. Distilled lessons can outperform raw traces on web and software tasks (Shen et al., 2026), continual consolidation can erode utility (Zhang et al., 2026), and ALFWorld and BabyAI can favor procedural insights over raw episodes (Hu et al., 2026). Our fixed-selection experiments connect these observations to cross-episode invariance: the same selected evidence can help or hurt depending on which parts compilation treats as transferable.

D.2 Data Selection, Valuation, and Teaching

Data selection is a central tool in model training and alignment. Dataset pruning targets informative or representative subsets (Sorscher et al., 2022). LIMA demonstrates strong alignment behavior from a small curated instruction set (Zhou et al., 2023). Textbooks Are All You Need shows the effect of highly curated synthetic corpora (Gunasekar et al., 2023). In skill induction, the selected object is a trajectory set whose members can jointly identify a procedure.

Demonstration selection methods optimize similarity, diversity, coverage, or learned retrieval (Liu et al., 2022; Rubin et al., 2022; Su et al., 2023; Zhang et al., 2022). Coreset and submodular methods preserve distributions or coverage (Sener & Savarese, 2018; Wei et al., 2015). Data valuation assigns influence to individual examples (Koh & Liang, 2017; Ghorbani & Zou, 2019). The support-set objective in Equation 3 preserves an induced rule. Its value can appear at higher interaction orders, so a completing set can carry functional value that is absent from independent per-item scores.

The exact-world support set also connects to teaching sets and teaching dimension (Mitchell, 1982; Goldman & Kearns, 1995; Zhu, 2015; Zhu et al., 2018). Classical machine teaching assumes a specified interpretation of teaching examples (Angluin, 1988). Agent trajectories add an interface layer because the same record can act as a logical constraint or as visible behavior. Proposition 6 formalizes this target-dependent interpretation.

D.3 In-Context Learning and Evidence Semantics

In-context learning can implement structured algorithms on synthetic function classes (Garg et al., 2022; Akyürek et al., 2023; Xie et al., 2022), depends on training-data structure (Chan et al., 2022), and exhibits induction-head copying (Olsson et al., 2022). Demonstration order, format, and label priors can materially change behavior (Lu et al., 2022; Zhao et al., 2021). These findings explain why a trajectory has two roles inside a target model: it states evidence about a hidden procedure and displays an action sequence. Our binding experiments measure the first role. The failure-copying and competence experiments measure how the second role changes the realized value of the same evidence.

D.4 Functional Interactions

Partial information decomposition distinguishes redundant, unique, and synergistic information (Williams & Beer, 2010). Harsanyi dividends and related interaction indices decompose joint value in cooperative games (Shapley, 1953; Grabisch & Roubens, 1999). We use the utility-level interaction spectrum to measure whether a functional skill is supported by combinations of experiences. The support-set analysis then asks how much of that jointly supported structure must be retained. The shortcut control connects the exact-world construction to shortcut learning (Geirhos et al., 2020). Compilation is also related to prompt compression (Jiang et al., 2023; Mu et al., 2023). Our closed-loop intervention adds a transfer criterion for skill induction: the compiled artifact should preserve structure that remains invariant across target episodes.