SCIQUOTIENT: FROM EXACT FAILURES TO TRANSFERABLE SCIENTIFIC STATE ABSTRACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific agents powered by large language models (LLMs) increasingly solve complex problems through long-horizon cycles of code generation, execution, and revision, accumulating failures that could serve as valuable training experience. Yet repairing a failure does not necessarily yield transferable knowledge, because the model may still misidentify which state distinctions matter. To make this gap measurable, we introduce SciQuotient, an integrated evaluation-and-training framework for learning transferable scientific state abstractions from exact failures. Given a fully specified finite dynamical system and a target observation, an LLM must construct the coarsest exact state abstraction that preserves all future target behavior. Because the correct abstraction can be certified exactly, SciQuotient uses certified quotient-preserving transformations to vary scientifically irrelevant aspects of the state and its representation while leaving the correct abstraction unchanged, exposing hidden abstraction failures that conventional held-out accuracy can miss. Across several model families, training on exact failures in a single representation strongly repairs the targeted behavior and supports partial transfer, but models trained this way remain brittle to mathematically equivalent re-encodings. In contrast, training on equivalent failures across multiple representations substantially broadens transfer to unseen representations and state configurations, with this transfer extending to held-out dynamical mechanisms without additional training. The resulting models can also surpass strong reference LLM baselines on SciQuotient transfer evaluations. Together, these results show that fixing known failures does not guarantee transfer, while representation-diverse exact-failure training helps models carry learned abstractions beyond the conditions in which those failures were observed.

1 INTRODUCTION

Scientific agents powered by large language models (LLMs) increasingly tackle long-horizon research workflows in which they hypothesize, write code, execute it, and revise (Yao et al., 2023; Boiko et al., 2023; Bran et al., 2024; Romera-Paredes et al., 2024; Lu et al., 2024). On realistic benchmarks such as SciCode (Tian et al., 2024), ScienceAgentBench (Chen et al., 2025), DiscoveryBench (Majumder et al., 2025), and MLAgentBench (Huang et al., 2024b), these agents still fail often, and their failures accumulate along the trajectory. A growing body of work therefore treats failure as supervision, through execution feedback, self-correction, and training on corrected trajectories (Le et al., 2022; Chen et al., 2024; Shinn et al., 2023; Madaan et al., 2023; Zelikman et al., 2022; An et al., 2023; Song et al., 2024; Kumar et al., 2025). What remains unclear is whether a corrected failure teaches the LLM something reusable or merely patches the observed instance.

We argue that for long-horizon scientific tasks this question is largely a question of *state abstraction*. When coding and tool use are otherwise adequate, an agent's decisions depend on which state distinctions it preserves; we isolate and measure this component exactly. Retaining too little merges situations that later diverge, so the error surfaces many steps after its cause; retaining too much fills the working context with detail that LLMs exploit unreliably (Liu et al., 2024). Deciding what to remember is thus a scientific judgment about which distinctions matter, and a repaired failure is valuable only if this judgment survives changes in how the problem is written. LLMs, however, are known to be sensitive to superficial reformulations of the same problem (Sclar et al., 2024; Wu et al.,

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.

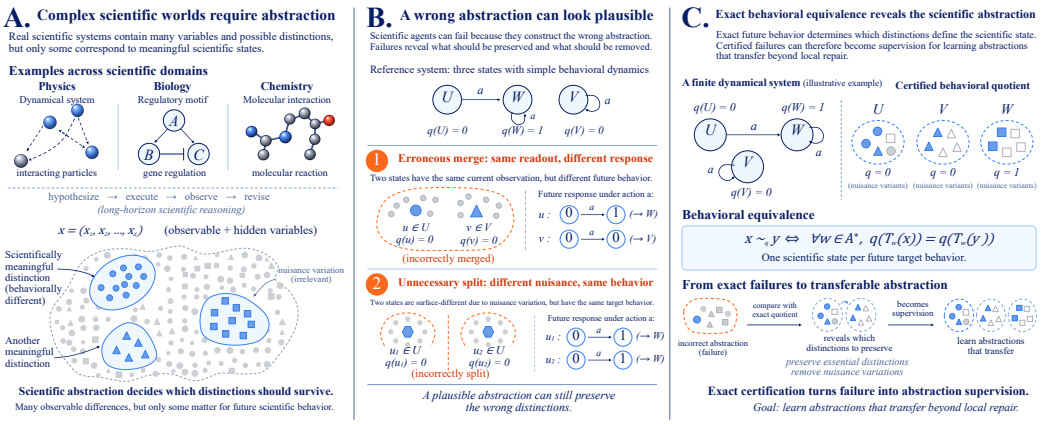

Figure 1: **Overview of SciQuotient. (A)** Long-horizon scientific systems expose many variables, but only some distinctions are scientifically meaningful; an abstraction decides which survive. **(B)** A plausible abstraction can be wrong in two ways: an *erroneous merge* identifies states with the same readout but different future responses, whereas an *unnecessary split* separates states that differ only in nuisance variation. **(C)** Future-behavioral equivalence $\sim_q$ defines one abstract state per future behavior. Comparing a failed abstraction with this certified quotient reveals which distinctions to keep or remove, turning exact failures into supervision that transfers beyond local repair.

2024; Mirzadeh et al., 2025; McCoy et al., 2024; Berglund et al., 2024), so accuracy on the repaired case says little about whether the abstraction itself was learned (Figure 1).

To make this measurable, we introduce **SciQuotient**, an integrated evaluation-and-training framework built on a task whose errors and solutions can be certified exactly. Given a fully specified finite dynamical system F and a target observation q, an LLM must write a symbolic expression $\hat{\Phi}$ defining the coarsest state abstraction that preserves all future target behavior, namely the behavioral quotient studied in automaton minimization and bisimulation (Hopcroft, 1971; Milner, 1989; Givan et al., 2003). The verifier compares the entire induced partition with the ground truth, so it detects both erroneous merges, which forget what matters, and unnecessary splits, which remember what does not. The same certificate plays three roles: it identifies failures, supplies the corrected target for training, and licenses quotient-preserving transformations, such as adding nuisance variables or re-encoding the state, that change the surface of a problem while provably leaving its answer unchanged. SciQuotient can therefore separate repairing an observed failure from learning an abstraction that remains valid under scientifically irrelevant variation.

Using SciQuotient, we compare two ways of turning the same exact failures into training data. Training on failures presented in a single representation repairs the targeted behavior completely and even transfers to unseen nuisance cardinalities and held-out mechanisms, yet the same model solves only 18.06% of instances once the state is re-encoded in a mathematically equivalent fused form. Presenting the same failure principles across equivalent representations closes this gap and carries it to mechanisms never used for training, raising held-out re-encoded accuracy from 9.72% to 75.00%. The pattern recurs from 2B to 32B parameters and across three further open model families, and on this interface the adapted 8B constructor matches or exceeds all nine unadapted reference LLMs on seven of eight conditions. Single-representation correction can remain formulation-bound, whereas diverse presentations improve transfer across equivalent formulations. The advantage is not uniform: it shrinks with scale and can reverse under familiar representations, yielding a broader but bounded profile rather than a universally stronger model.

Our contributions are fourfold:

- We introduce SciQuotient, which turns the question of whether failure correction becomes reusable knowledge into an exactly certifiable problem of scientific state abstraction, and uses one certificate for both training and evaluation.
- We show that near-perfect repair can coexist with severe brittleness under equivalent reformulation, a gap that post-training accuracy conceals.

- We demonstrate that representation-diverse failure training substantially broadens transfer across equivalent encodings, nuisance cardinalities, and held-out mechanisms.
- We chart this transfer across four model scales, four model families, and nine unadapted reference LLMs, identifying where the advantage weakens or reverses.

2 PRELIMINARIES

2.1 FINITE DYNAMICAL SYSTEMS

We consider finite deterministic dynamical systems $F = (\mathcal{X}, \mathcal{A}, \{T_a\}_{a\in\mathcal{A}})$, where $\mathcal{X}$ is a finite state space, $\mathcal{A}$ a finite action set, and each action a induces a deterministic transition $T_a : \mathcal{X} \to \mathcal{X}$. An action sequence $w = (a_1, \ldots, a_k) \in \mathcal{A}^*$ acts by composition, $T_w = T_{a_k} \circ \cdots \circ T_{a_1}$, with T_ϵ the identity for the empty sequence. Actions model interventions an agent may apply, such as perturbing a variable, advancing a simulation, or running an assay, and w is a finite experimental protocol.

The scientific task does not ask for the full state but specifies a target observation $q : \mathcal{X} \to \mathcal{Y}$, the quantity whose behavior is of interest. What matters about a state x is therefore not only its current readout $q(x)$ but the entire family of future readouts $q(T_w(x))$ over all protocols $w \in \mathcal{A}^*$.

2.2 FUTURE-BEHAVIORAL EQUIVALENCE

Two states $x, y \in \mathcal{X}$ are future-behaviorally equivalent if no protocol separates them via the target:

$$x \sim_q y \quad \Longleftrightarrow \quad q(T_w(x)) = q(T_w(y)) \ \text{ for all } w \in \mathcal{A}^*. \tag{1}$$

The relation $\sim_q$ partitions $\mathcal{X}$ into classes indistinguishable by any future target behavior. Equal current readouts are not sufficient: two states with $q(x) = q(y)$ may differ in a latent variable that makes their readouts diverge after some protocol. For instance, two cultures with the same fluorescence readout but different states of an unobserved regulator are not equivalent if some induction protocol later separates their fluorescence, whereas a sample label that no protocol can affect carries no target information. Any distinction that never affects $q(T_w(\cdot))$ is irrelevant and can be forgotten.

2.3 EXACT STATE ABSTRACTION AND BEHAVIORAL QUOTIENTS

A state abstraction $\Phi : \mathcal{X} \to \mathcal{Z}$ maps each state to an abstract state and induces the relation $x \sim_\Phi y \iff \Phi(x) = \Phi(y)$, that is, a partition of $\mathcal{X}$. We call Φ exact with respect to q when $\sim_\Phi = \sim_q$. Exactness rules out two complementary errors: Φ is too coarse if it merges states with different future target behavior, forgetting information needed to predict q, and too fine if it separates equivalent states, remembering distinctions the target never uses. Keeping the full state therefore preserves all predictive information yet is generally not exact, since it splits equivalent states. The classes of $\sim_q$ form the unique coarsest exact partition, and we call $\mathcal{X}/\sim_q$ the **behavioral quotient**. It is the finite, deterministic counterpart of automaton minimization, bisimulation, and model minimization in Markov decision processes (Milner, 1989; Givan et al., 2003; Li et al., 2006), retaining exactly the distinctions that matter for future target behavior. We use this classical structure as an exact reference for evaluating and training LLMs, not as a new equivalence relation.

3 SCIQUOTIENT

SciQuotient turns the quotient of Section 2 into a loop in which one exact certificate serves evaluation and training alike: it scores an LLM's abstraction, converts each failure into a corrected training target, and generates equivalent variants on which repair and transfer can be told apart.

3.1 TASK INTERFACE

Each instance gives the LLM a complete specification of (F, q) and asks for a single symbolic expression, $(F, q) \mapsto \hat{\Phi}$. We call such a model an abstraction constructor. Construction is one-shot, with no search, execution, verifier feedback, or revision, so the output directly reflects which distinctions the LLM believes it must remember. Outputs are scored by the partition they induce rather than by string match, so syntactically different expressions are equivalent whenever they group states identically, which matters under the re-encodings below.

3.2 Exact Certification

A prediction $\hat{\Phi}$ induces $\sim_{\hat{\Phi}}$, and correctness is whole-partition equality with the ground truth, $\sim_{\hat{\Phi}} \stackrel{?}{=} \sim_q$, checked over the complete finite state space. An expression that fails to define a partition of $\mathcal{X}$ is invalid and counted as incorrect. For a valid prediction, a mismatch takes two forms, which can co-occur. An erroneous merge groups states whose future q-behavior differs and so forgets information needed for prediction; an unnecessary split separates equivalent states and so remembers distinctions the coarsest abstraction should discard. A prediction is exact only when neither occurs. Since $\mathcal{X}$ is finite, the reference $\sim_q$ is computed exactly by partition refinement (Hopcroft, 1971; Paige & Tarjan, 1987), which splits classes of equal readout until all action successors agree and terminates at $\sim_q$ (Appendix A). Because the check is exhaustive rather than a heuristic or LLM judge, every failure is certified, its type is known, and the correct partition serves as a training target.

3.3 Certified Quotient-Preserving Transformations

Certification also lets us vary an instance without changing its answer. A transformation mapping (F, q) to (F', q') is quotient-preserving if the two behavioral quotients are isomorphic (Figure 2).

Nuisance expansion augments each state $x \in \mathcal{X}$ with a variable $n \in \mathcal{N}$ that is invisible to the target, $q'(x, n) = q(x)$, and never feeds back into the original dynamics, $T'_a(x, n) = (T_a(x), g_a(x, n))$, where g_a updates only the nuisance component. Then $(x, n) \sim_{q'} (y, m)$ exactly when $x \sim_q y$, so states differing only in n must be merged; observation independence alone would not suffice, since a nuisance that steered T_a could later change the target. We vary the nuisance cardinality $|\mathcal{N}|$ through the cyclic groups $\mathbb{Z}_2$, $\mathbb{Z}_3$, and $\mathbb{Z}_4$. Representation re-encoding rewrites the same system through a bijection $\rho : \mathcal{X} \to \mathcal{X}'$, with $T'_a = \rho \circ T_a \circ \rho^{-1}$ and $q' = q \circ \rho^{-1}$, so that $\rho(x) \sim_{q'} \rho(y)$ exactly when $x \sim_q y$ and the quotient is preserved up to the known correspondence ρ. We use it to map a factorized encoding, in which state variables appear separately, to a fused encoding, in which they are packed into a single composite variable. Both transformations preserve the partition structure, though not the text of the target expression (proofs in Appendix A.2), so a failure on a transformed instance reflects sensitivity to irrelevant state or notation, not a change in the abstraction to be recovered.

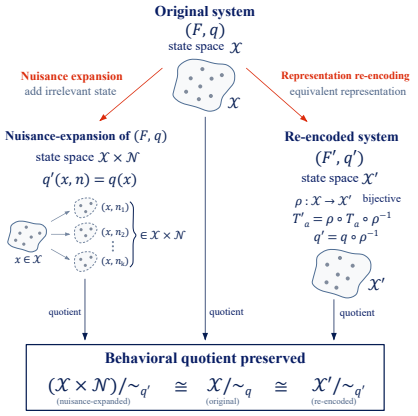

Figure 2: **Quotient-preserving transformations.** Nuisance expansion adds irrelevant state, and re-encoding applies a bijection ρ; both change the surface of (F, q) while the behavioral quotient stays isomorphic.

3.4 Learning from Exact Failures

For every instance on which a constructor is non-exact, certification supplies an expression $\Phi^\star$ with $\sim_{\Phi^\star} = \sim_q$, so $(F, q) \mapsto \Phi^\star$ becomes a supervised example whose correction comes from the exact quotient rather than heuristic or model-generated feedback. **Single-representation training** (S) presents each corrected failure in one representation; **representation-diverse training** (D) presents the same failures also through re-encoded variants (F', q') of Section 3.3, with the target transported by ρ, $\Phi^{\star\prime} = \Phi^\star \circ \rho^{-1}$, which induces exactly the re-encoded quotient. S and D thus share the failure principles and differ mainly in whether each is seen in one form or several (Section 4.2).

3.5 Repair and Transfer Evaluation

After training, each constructor is evaluated without further adaptation by the same certificate, counting a prediction as correct only if $\sim_{\hat{\Phi}} = \sim_q$; the verifier scores the output after generation and never returns feedback. We separate failure repair, exactness under the conditions that produced the supervision, from transfer, exactness under conditions never used for correction. Transfer is probed along the two quotient-preserving axes, unseen nuisance cardinalities and re-encoded representations, and along a third axis that does not preserve any particular quotient: dynamical mechanisms

held out from adaptation, which test whether the learned rule extends to new system structure. To diagnose re-encoding failures we additionally pair the predictions for an instance i on its factorized form, $\hat{\Phi}_i^U$, and on its fused form obtained through ρ_i, $\hat{\Phi}_i^F$, and record the *representation consistency* $C_i = \mathbf{1}\big[\sim_{\hat{\Phi}_i^U} = (\rho_i \times \rho_i)^{-1}(\sim_{\hat{\Phi}_i^F})\big]$, where ρ_i aligns the two state spaces so that the partitions can be compared state by state. A pair is *consistent* if the two partitions correspond under ρ_i, *inconsistent* if both are valid but disagree, and *invalid* if at least one output fails to define a partition. Consistency is a diagnostic of representation sensitivity, not a substitute for exactness: two exact predictions are always consistent, but two identically wrong ones are consistent as well (Appendix A.4).

This design places SciQuotient between a benchmark, which counts failures without testing whether correcting them teaches anything, and a training recipe, which, evaluated in-distribution, cannot test the invariance it should provide. Because one certificate defines the error, correction, and equivalent variants, a drop under a shift reflects what the LLM learned, not label noise or a changed target.

4 Experiments

4.1 Experimental Setup

Model and evaluation conditions. We use Qwen3-VL-8B-Instruct (Bai et al., 2025) as the primary constructor, adapted under S and D (Section 4.2). Training uses the canonical nuisance cardinality $\mathbb{Z}_3$, holding out $\mathbb{Z}_2$ and $\mathbb{Z}_4$. Crossing encodings with cardinalities yields four panels on adaptation-seen mechanisms: E1 (factorized, $\mathbb{Z}_3$) measures repair, E2 (fused, $\mathbb{Z}_3$) re-encoding transfer, E3 (factorized, $\mathbb{Z}_2, \mathbb{Z}_4$) cardinality transfer, and E4 (fused, $\mathbb{Z}_2, \mathbb{Z}_4$) both jointly. H1–H4 apply them to mechanisms absent from all adaptation data, so H1 isolates mechanism transfer and H4 combines all three shifts. Only E1 matches the representation and cardinality seen by S; D has also seen fused encodings of adaptation mechanisms, never held-out cardinalities or mechanisms. Source tasks and mechanisms are split before generating quotient-preserving variants, preventing equivalent realizations of the same base system from crossing adaptation and evaluation splits.

Evaluation and supporting models. All checkpoints are evaluated by greedy one-shot generation, and we report Exact accuracy, the percentage of instances with $\sim_{\hat{\Phi}} = \sim_q$. Results pool three training seeds; D−S differences are in percentage points with 95% paired bootstrap intervals over task instances (Efron & Tibshirani, 1993). To test whether the S–D contrast is specific to one model, we repeat both conditions at 2B, 4B, and 32B and on three independently pretrained families, Llama-3.1-8B (Grattafiori et al., 2024), OLMo-3-7B (Team Olmo et al., 2025), and Gemma-4-12B (Gemma Team et al., 2026), with unadapted reference LLMs as one-shot difficulty references.

4.2 Constructing Certified Failure Experience

We build the adaptation data so that every supervision signal is a mathematical fact rather than a judgment, and so that S and D differ mainly in how the same facts are written.

Failure mining and formalization. We start from public scientific workloads such as SciCode (Tian et al., 2024), ScienceWorld (Wang et al., 2022), and ScienceAgentBench (Chen et al., 2025), together with internally designed long-horizon scientific tasks. Multiple heterogeneous LLM generators produce candidate execution trajectories; the failure-mining pool has included DeepSeek, Kimi, GLM, and Grok models. Because agents fail for many reasons, a trajectory proceeds only if its failure can be localized to a wrong state distinction. Around the erroneous decision point we formalize a local finite system $F = (\mathcal{X}, \mathcal{A}, \{T_a\})$: $\mathcal{X}$ holds the finitely many states relevant to later decisions, $\mathcal{A}$ the executable subsequent interventions, each T_a comes from dynamics known in the task or environment, and q is the scientific quantity the decision must ultimately preserve.

Exact certification. On each system we compute $x \sim_q y$, that is, $q(T_w(x)) = q(T_w(y))$ for every $w \in \mathcal{A}^*$. Finite partition refinement returns the unique behavioral quotient $X/\sim_q$, which is serialized under the fixed output grammar into a canonical symbolic target $\Phi^\star$ satisfying $\sim_{\Phi^\star} = \sim_q$ (Appendix A). The trajectory therefore determines only the provenance of a failure and the decision context to formalize, while the correct supervision is fixed independently by the verifier, so no generator can pass its own error into the label.

Table 1: Repair and transfer under evaluation shifts (Qwen3-VL-8B-Instruct). Exact accuracy (%) of single-representation (S) and representation-diverse (D) failure training on adaptation-seen (E1–E4) and held-out (H1–H4) mechanisms. D−S is the paired difference in percentage points, with 95% intervals from a paired task-instance bootstrap over the pooled three-seed evaluation set.

Panel	Shift	S	D	D−S	95% CI
E1	Repair	100.00	100.00	+0.00	[0.00, 0.00]
E2	Re-encoding	18.06	100.00	+81.94	[77.43, 86.11]
E3	Cardinality	100.00	100.00	+0.00	[0.00, 0.00]
E4	Re-encoding + cardinality	19.10	99.65	+80.56	[77.08, 83.68]
H1	Mechanism	95.83	96.53	+0.69	[-3.47, 4.86]
H2	Mechanism + re-encoding	9.72	75.00	+65.28	[57.64, 72.92]
H3	Mechanism + cardinality	90.62	90.97	+0.35	[-3.47, 4.17]
H4	Mechanism + re-encoding + cardinality	13.54	74.31	+60.76	[54.51, 66.67]

Equivalent representation construction. For each certified failure, a known bijection $\rho : \mathcal{X} \to \mathcal{X}'$ gives $T'_a = \rho \circ T_a \circ \rho^{-1}$ and $q' = q \circ \rho^{-1}$, so the original and re-encoded systems have isomorphic behavioral quotients (Section 3.3). Factorized and fused inputs are two concrete realizations of this construction rather than free-text rewrites, so their equivalence is proven, not hoped for.

Controlled adaptation. After filtering, formalization, and exact certification, we obtain about 10,000 verifier-certified adaptation records. S and D use matched budgets drawn from the same certified failure pool. S and D draw on the same certified failure principles: S uses only the canonical factorized representation, whereas D allocates a matched data budget across factorized and fused realizations. Parent checkpoint, record budget, replay, optimizer updates, seed set, and training hyperparameters are matched, so the main experimental difference is the representation composition of the certified failure experience.

4.3 From Failure Repair to Transferable Abstraction

We first ask what single-representation training actually repairs. Measured on the conditions that generated the supervision, the answer is everything: both S and D reach 100.00% on E1 (Table 1). S also transfers along two axes it never trained on, remaining exact under unseen nuisance cardinalities (E3, 100.00%) and largely exact on held-out mechanisms (H1, 95.83%; H3, 90.62%). By the usual standard of held-out accuracy, the corrected failures appear to have taught S the abstraction.

Re-encoding overturns this conclusion. When the same instances are written in the fused encoding, whose behavioral quotient is identical by construction, S falls to 18.06% on E2 and 19.10% on E4, whereas D stays at 100.00% and 99.65%, gaps of +81.94 and +80.56 points. Because the behavioral quotient and the target are unchanged, the drop cannot be attributed to a change in the abstraction that should be recovered; it isolates sensitivity to representation. Figure 3 shows what S does instead. Pairing each factorized prediction with its fused counterpart, only 4.5% of S's pairs on E1 and E2

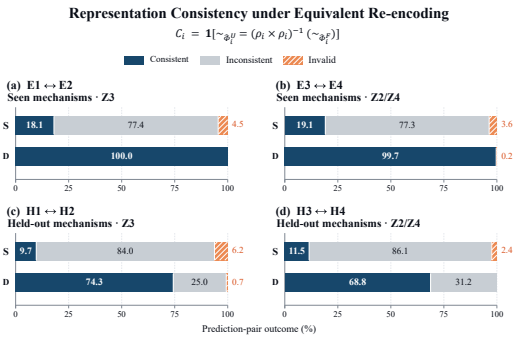

Figure 3: **Representation consistency of paired predictions.** Each factorized prediction is compared with its fused counterpart through C_i: consistent (partitions correspond under ρ_i), inconsistent (both valid but disagree), or invalid (at least one fails to define a partition). S is mostly valid yet inconsistent, so its errors are abstraction errors rather than parsing failures, whereas D stays largely consistent. Consistency is a diagnostic and does not imply exactness.

Table 2: Generalization across model scale. Exact accuracy (%) of Qwen3-VL constructors from 2B to 32B parameters, reported as S/D. Performance on the factorized panels (E1, E3) is near saturation at every size, while the gap between S and D on re-encoded panels (E2, E4, H2, H4) narrows as S acquires re-encoding transfer at 32B.

Model Size	**Adaptation-seen mechanisms**				**Held-out mechanisms**			
	E1 *S/D*	**E2** *S/D*	**E3** *S/D*	**E4** *S/D*	**H1** *S/D*	**H2** *S/D*	**H3** *S/D*	**H4** *S/D*
2B	100.00/100.00	0.00/100.00	99.83/99.65	0.00/99.48	86.11/90.28	0.00/46.53	84.38/89.24	0.00/46.18
4B	99.65/99.31	15.97/100.00	98.96/98.78	22.92/99.83	75.00/78.47	0.69/65.97	76.04/79.51	5.21/64.93
8B	100.00/100.00	18.06/100.00	100.00/100.00	19.10/99.65	95.83/96.53	9.72/75.00	90.62/90.97	13.54/74.31
32B	100.00/100.00	93.75/100.00	100.00/100.00	93.58/100.00	90.67/95.83	88.89/87.50	91.32/91.67	78.47/87.50

Table 3: Cross-family generalization and unadapted reference LLMs. Exact accuracy (%). Adapted open model families report S/D. Unadapted reference LLMs are evaluated one-shot and serve as difficulty references rather than matched training comparisons.

Model	**Adaptation-seen mechanisms**				**Held-out mechanisms**			
	E1	**E2**	**E3**	**E4**	**H1**	**H2**	**H3**	**H4**
Unadapted reference LLMs								
GPT-5.6 Luna	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DeepSeek-V4.1-Flash	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kimi-K3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.04
Qwen3.8-Max-0902	0.00	0.00	0.00	1.04	0.00	0.00	0.00	0.00
GPT-5.6 Terra	12.50	0.00	19.79	0.00	10.42	0.00	9.38	0.00
Grok-4.6	37.50	14.58	28.65	15.10	22.92	12.50	29.17	13.54
Claude Sonnet 5	78.13	34.38	74.48	71.88	68.75	51.04	69.27	57.29
GPT-5.6 Sol	69.79	60.42	67.19	62.50	81.25	60.42	80.21	69.79
Claude Opus 5	93.75	93.75	84.90	91.15	66.67	75.00	78.12	77.08
SciQuotient-adapted open model families (S/D)								
Llama-3.1-8B	100.00/98.26	7.99/87.85	100.00/98.44	7.64/91.67	86.81/76.39	4.17/69.44	87.15/74.31	3.47/67.01
OLMo-3-7B	100.00/100.00	0.00/98.26	100.00/100.00	0.17/96.53	93.06/95.83	0.00/41.67	95.14/94.44	0.00/36.81
Gemma-4-12B	81.94/67.71	2.43/51.39	81.60/73.26	7.29/49.48	68.06/68.06	11.11/40.28	65.97/67.71	11.81/38.54

contain an invalid output, while 77.4% are valid partitions that disagree with S's own factorized answer. S therefore produces well-formed but representation-dependent abstractions: for the same underlying system, its factorized and fused predictions frequently induce different partitions. This pattern is consistent with reliance on representation-specific cues, although the behavioral test does not identify the model's internal strategy (Geirhos et al., 2020). Such a model can pass same-representation evaluations yet fail when the same system is expressed in another notation.

Since D trains on fused encodings of adaptation mechanisms, its E2 score shows that the gap can be closed, not yet that the rule transfers. The decisive test is the held-out mechanisms, which D has never seen in any representation. There D reaches 75.00% on H2 and 74.31% on H4, compared with 9.72% and 13.54% for S, gains of $+65.28$ $[57.64, 72.92]$ and $+60.76$ $[54.51, 66.67]$ points. The remaining errors are again almost entirely valid but inconsistent partitions (25.0% and 31.2% of pairs, with at most 0.7% invalid), indicating abstraction errors on harder instances rather than formatting failures. In contrast, where the representation is familiar, the paired intervals for D and S include zero (H1, $+0.69$ $[-3.47, 4.86]$; H3, $+0.35$ $[-3.47, 4.17]$). Diversity therefore does not make the constructor better at mechanisms in general; it makes the correction far less dependent on notation, which lets it travel to new mechanisms when the notation also changes. This mirrors domain randomization and invariant learning (Tobin et al., 2017; Arjovsky et al., 2019): seeing one failure principle under several equivalent surfaces makes future behavior a representation-invariant target for predicting the correct partition. We next ask whether scale supplies this invariance.

4.4 Generalization Across Model Scale

If S fails because a small model binds corrections to notation, a larger parent model with a more representation-invariant prior might make diversity unnecessary. Table 2 repeats both conditions from 2B to 32B parameters. E1 and E3 nearly saturate at every scale, with both conditions at or

above 98.78% for both conditions, so capacity is never the bottleneck for fixing the observed failures. Re-encoding transfer under S, however, emerges only with scale: S reaches 0.00%, 15.97%, 18.06%, and 93.75% on E2 and 0.00%, 0.69%, 9.72%, and 88.89% on H2 as size grows from 2B to 32B. D reaches 100.00% on E2 at every size, and its held-out re-encoded accuracy rises steadily, from 46.53% to 65.97%, 75.00%, and 87.50% on H2, with nearly identical values on H4.

Two consequences follow. First, diverse failure experience supplies an invariance that otherwise appears only at substantially larger scale: the 2B D constructor outperforms the S constructor four times its size on every re-encoded panel (E2, 100.00% vs. 18.06%; H4, 46.18% vs. 13.54%). Second, the benefit of diversity is largest precisely where the parent model lacks invariance. The H2 gap is $+46.53$ points at 2B and $+65.28$ at both 4B and 8B, then disappears at 32B, where S slightly exceeds D (88.89% vs. 87.50%, -1.39) because single-representation corrections already largely carry across encodings; this point estimate is an exception to a uniform D advantage, not a statistically resolved reversal. Even there, D still improves H4 by $+9.03$ and H1 by $+5.16$ points, so the advantage narrows rather than disappears. Mechanism transfer is also not monotone in size: the 4B model is weaker than the 2B model on H1 under both conditions (75.00% and 78.47% vs. 86.11% and 90.28%), so properties of the parent checkpoint beyond parameter count matter. Because all four sizes share one pretraining lineage, we next test independently pretrained families.

4.5 Cross-Family Generalization and Unadapted Reference LLMs

Table 3 applies S and D unchanged to Llama-3.1-8B, OLMo-3-7B, and Gemma-4-12B. The central separation replicates in every family, so it is not unique to the Qwen checkpoint: S remains near zero on re-encoded panels (E2, 7.99%, 0.00%, and 2.43%; H2, 4.17%, 0.00%, and 11.11%), and D raises these to 87.85%, 98.26%, and 51.39% on E2 and to 69.44%, 41.67%, and 40.28% on H2.

The families also delineate the limits of the effect. OLMo-3-7B learns near-perfect invariance on adaptation mechanisms (98.26% on E2) yet carries much less of it to held-out mechanisms (41.67% on H2), so representation invariance and mechanism generalization are separable abilities. For Llama-3.1-8B, D costs 10 to 13 points on familiar held-out panels (H1, 86.81% to 76.39%; H3, 87.15% to 74.31%), and for Gemma-4-12B, repair itself is incomplete under both conditions, with D below S on E1 (67.71% vs. 81.94%) and E3 (73.26% vs. 81.60%). At matched volume D devotes a smaller share of its budget to any single representation, so some parent models may trade familiar-form accuracy for broader invariance; our results do not isolate whether this tradeoff arises from coverage allocation, optimization, or pretrained inductive biases. Diversity thus consistently improves re-encoding transfer, while its effect on familiar conditions depends on the parent.

One-shot unadapted reference LLMs under the same interface place these numbers in context. Exact whole-partition construction is demanding even at the frontier: four of the nine references score at most 1.04% on every panel, and even the strongest, Claude Opus 5, falls from 93.75% on E1 to 66.67% on H1. Frontier LLMs are also not immune to notation, as Claude Sonnet 5 drops from 78.13% on E1 to 34.38% on E2. Against this reference set, the adapted Qwen3-VL-8B D constructor matches or exceeds every unadapted reference LLM on seven of eight panels, trailing only Claude Opus 5 on H4 (74.31% vs. 77.08%), and the 32B D constructor exceeds all of them on every panel. These comparisons are unmatched in adaptation history and inference configuration and do not rank broader scientific-agent capability, but show that certified, representation-diverse failure experience yields specialized performance that general-purpose post-training does not reach.

Taken together, the results show that certified corrections can reliably repair observed failures across scales and in three of the four model families, but successful correction alone does not guarantee representation-stable transfer. Single-representation supervision remains strongly representation-dependent, whereas training on certified equivalent variants substantially reduces this dependence across representations, nuisance cardinalities, and mechanisms. This distinction separates local failure repair from learning a more transferable scientific abstraction.

5 Related Work

LLM scientific agents and learning from failure. LLM agents now plan, execute code, and revise across multi-step research tasks (Yao et al., 2023; Boiko et al., 2023; Bran et al., 2024; Romera-Paredes et al., 2024; Lu et al., 2024), and scientific benchmarks show how often they still

fail (Tian et al., 2024; Chen et al., 2025; Majumder et al., 2025; Huang et al., 2024b). Failures have been turned into signal through execution feedback (Le et al., 2022; Chen et al., 2024), verbal reflection and iterative refinement (Shinn et al., 2023; Madaan et al., 2023), process supervision (Lightman et al., 2024), and fine-tuning on corrected or contrasted trajectories (Zelikman et al., 2022; An et al., 2023; Song et al., 2024; Zeng et al., 2024; Kumar et al., 2025), while memory architectures decide which experience an agent retains (Packer et al., 2023; Park et al., 2023; Wang et al., 2024; Zhao et al., 2024). These works measure gains on the repaired task or its distribution, and self-correction without external signals remains unreliable (Huang et al., 2024a); whether a correction survives reformulation is left untested, which SciQuotient tests with exact certificates.

Robustness of LLMs to equivalent reformulations. LLM behavior shifts under changes that leave a task's meaning intact, including prompt formatting (Sclar et al., 2024), counterfactual variants of familiar tasks (Wu et al., 2024), symbolic templates of math problems (Mirzadeh et al., 2025), the frequency of output forms (McCoy et al., 2024), reversed relations (Berglund et al., 2024), and compositional recombination (Lake & Baroni, 2018; Dziri et al., 2023), and generated answers are often inconsistent with a model's own judgments (Li et al., 2024), often traced to shortcut learning (Geirhos et al., 2020). These studies diagnose brittleness in pretrained models but rarely certify that two variants share one answer or examine what failure training does to it. Our transformations provide such certificates and show that repair leaves brittleness that diversity largely removes.

State abstraction and behavioral equivalence. Deciding which state distinctions to keep is classical in automaton minimization and bisimulation (Hopcroft, 1971; Milner, 1989), model minimization and state abstraction for Markov decision processes (Givan et al., 2003; Li et al., 2006; Ferns et al., 2004), bisimulation-based representation learning (Zhang et al., 2021), and active automaton learning (Angluin, 1987). These methods compute or learn abstractions for a fixed system but have not been used to supervise or evaluate LLMs that construct them from a description. Metamorphic testing checks relations between outputs on transformed inputs (Chen et al., 2018), but consistency alone misses a model that errs identically in both forms. SciQuotient reuses the exact theory as a certificate: each transformation carries a proven invariant and the full quotient is known, so one object defines correctness, generates targets, and licenses transfer tests.

6 Limitations

SciQuotient isolates state abstraction in finite, deterministic systems with fully specified dynamics, trading realism for exactness: continuous, stochastic, and partially observed systems, and dynamics that must be inferred from experiments, lie outside its scope, exhaustive certification limits scalability, and translating real scientific failures into finite systems requires additional modeling choices. Construction is one-shot, leaving tool use, memory, and revision open. We test two transformation families, and since D sees fused encodings during training, generalization to arbitrary unseen representation families is not established. Exactness concerns output partitions, not the internal algorithm, and credits no partial progress. Matched volume does not disentangle every effect of coverage and optimization, and the pooled task-instance bootstrap does not characterize training-seed uncertainty, so scale and family reversals are reported descriptively, and unadapted reference LLMs are unmatched.

7 Conclusion

LLM-based scientific agents will keep failing on long-horizon tasks, where deciding which state distinctions to preserve is an important component of reliable reasoning. SciQuotient makes this judgment exactly measurable and trainable: one certificate identifies abstraction failures, converts them into supervision, and generates equivalent variants that separate repair from transfer. Correcting failures in a single representation repairs them almost perfectly yet yields little transfer under re-encoding, whereas presenting them across equivalent representations produces abstractions that transfer more reliably to held-out re-encoded instances, unseen nuisance cardinalities, and held-out mechanisms, with an advantage that narrows at scale and can reverse on familiar encodings. Learning from failure thus depends not only on which failures are corrected but on whether they are seen in equivalent forms. Extending the certificate to stochastic, continuous systems and full research loops is the next step; it already shows that successful repair is weak evidence of reusable abstraction.

AI USE STATEMENT

We used generative AI tools in two ways: to generate synthetic candidate scientific trajectories from which abstraction failures were mined, with every training target admitted only after exact certification by our verifier, and to polish the language of the manuscript. Exact abstraction labels and all reported evaluation outcomes are determined by the mathematical certification procedure rather than by an AI judge. Research ideas, experimental design, analysis, and all scientific claims are our own. We have reviewed all AI-assisted content and take responsibility for the final content of this work.

ETHICS STATEMENT

This work does not involve human subjects, personal data, sensitive attributes, or high-stakes real-world decision making. SciQuotient uses synthetic finite dynamical systems and public scientific task contexts; LLM-generated trajectories serve only as candidate sources of abstraction failures, while all training targets and evaluation outcomes are determined by exact mathematical certification.

REPRODUCIBILITY STATEMENT

The formal task definition, exact certification procedure, quotient-preserving transformations, failure-data construction, and evaluation protocol are specified in Sections 2–4. Section 4 describes the experimental splits, S/D construction and matching controls, model families, evaluation panels, seed aggregation, and paired-bootstrap analysis. Appendix A provides the certification procedure and proofs of the transformation guarantees. All abstraction targets and reported outcomes are determined by exact finite-state certification.

REFERENCES

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Learning from mistakes makes LLM better reasoner. *arXiv preprint arXiv:2310.20689*, 2023.

Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "A is B" fail to learn "B is A". In *The Twelfth International Conference on Learning Representations*, 2024.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys*, 51(1):4:1–4:27, 2018.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*, 2024.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, pp. 70293–70332, 2023.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Norman Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, pp. 162–169, 2004.

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.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1–2):163–223, 2003.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

John Hopcroft. An $n \log n$ algorithm for minimizing states in a finite automaton. In Zvi Kohavi and Azaria Paz (eds.), *Theory of Machines and Computations*, pp. 189–196. Academic Press, 1971.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*, 2024a.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309, 2024b.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2873–2882, 2018.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21314–21328, 2022.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006.

Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. Benchmarking and improving generator-validator consistency of language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

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, pp. 46534–46594, 2023.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121, 2024.

Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

Joon Sung Park, Joseph C. O'Brien, Carrie J. 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*, pp. 1–22, 2023.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2024.

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, pp. 8634–8652, 2023.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7584–7600, 2024.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. SciCode: A research coding benchmark curated by scientists. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30624–30650, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

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.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022. doi: 10.18653/v1/2022.emnlp-main.775.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1819–1862, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488, 2022.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *The Ninth International Conference on Learning Representations*, 2021.

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.

A Exact Certification and Transformation Guarantees

A.1 A Finite Characterization of All Future Behavior

Let $R_0 = \{(x, y) : q(x) = q(y)\}$, and recursively define

$$R_{k+1} = \{(x, y) \in R_0 : (T_a(x), T_a(y)) \in R_k \text{ for every } a \in \mathcal{A}\}. \tag{2}$$

For every $k \geq 0$, membership in R_k is equivalent to matching target observations under all action sequences of length at most k. The base case is the empty sequence. Inductively, a nonempty sequence is an action a followed by a sequence of length at most k, and matching observations is exactly the successor condition in Equation (2), together with equality at the current state.

Each R_k is an equivalence relation and $R_{k+1} \subseteq R_k$, so the associated partitions only split. Because $\mathcal{X}$ is finite, refinement reaches some K with $R_{K+1} = R_K$. Matching observations at every finite horizon is exactly the definition of $\sim_q$, hence $R_K = \sim_q$. This yields a terminating certificate for an unbounded requirement without enumerating action sequences.

A candidate expression is evaluated on the full state space. If it is not defined on that space, it does not define a partition and is counted as incorrect. Otherwise, equality of its output values induces a partition, which is compared with R_K independently of abstract label names. An erroneous merge has $\hat{\Phi}(x) = \hat{\Phi}(y)$ but $(x, y) \notin R_K$; an unnecessary split has $(x, y) \in R_K$ but $\hat{\Phi}(x) \neq \hat{\Phi}(y)$. Both types can occur in one prediction, so ruling out only one does not establish exactness.

A.2 Why the Transformations Preserve the Quotient

Nuisance expansion. Let $\pi(x, n) = x$. The conditions in Section 3.3 give $\pi \circ T'_a = T_a \circ \pi$ and $q' = q \circ \pi$. Induction over actions yields $\pi \circ T'_w = T_w \circ \pi$, hence

$$q'(T'_w(x, n)) = q(T_w(x)) \tag{3}$$

for every n and every finite w. Thus $(x, n) \sim_{q'} (y, m)$ if and only if $x \sim_q y$, and each original behavioral class corresponds to one expanded class regardless of nuisance cardinality. A valid transported target is $\Phi^{\star\prime}(x, n) = \Phi^\star(x)$. The proof needs both observation independence and the absence of nuisance feedback into the original dynamics: if n could influence T_a, two states with equal x could later produce different target observations.

Bijective re-encoding. If $T'_a = \rho \circ T_a \circ \rho^{-1}$, intermediate inverse maps cancel along a sequence, giving $T'_w = \rho \circ T_w \circ \rho^{-1}$. Therefore

$$q'(T'_w(\rho(x))) = q(T_w(x)), \tag{4}$$

and $\rho(x) \sim_{q'} \rho(y)$ exactly when $x \sim_q y$. The induced map $[x]_q \mapsto [\rho(x)]_{q'}$ is a bijection between quotient classes that preserves their labeled transitions, and the transported target $\Phi^{\star\prime} = \Phi^\star \circ \rho^{-1}$ induces the correct partition. Composing the two transformations retains the same guarantee.

A.3 A Concrete Distinction Between Merging and Splitting

Consider three states u, v, w with one action a, transitions $T_a(u) = w$, $T_a(v) = v$, and $T_a(w) = w$, and observations $q(u) = q(v) = 0$, $q(w) = 1$. Grouping u with v by their current observation is an erroneous merge: after one action their observations become 1 and 0. The exact quotient therefore has three classes. If an independent nuisance coordinate is added, all copies of u belong together, as do all copies of v and all copies of w, and keeping the nuisance coordinate in the abstraction unnecessarily splits these classes. The target is neither the current readout alone nor the complete concrete state, but exactly the distinctions required by future observations.

A.4 Representation Consistency Versus Exactness

For corresponding factorized and fused instances, write $R_i^U = \sim_{\hat{\Phi}_i^U}$ and $R_i^F = \sim_{\hat{\Phi}_i^F}$ for the two predicted relations, and let ρ_i map factorized states to fused states. When both predictions are valid, consistency is

$$C_i = \mathbf{1}\left[R_i^U = (\rho_i \times \rho_i)^{-1}(R_i^F)\right], \tag{5}$$

where (x, y) belongs to the inverse image exactly when $(\rho_i(x), \rho_i(y)) \in R_i^F$. Invalid predictions are reported separately in Figure 3. Two exact predictions must be consistent, because the reference quotients correspond under ρ_i (Appendix A.2). The converse is false: predicting a single abstract state in both encodings is consistent even when the correct quotient has several classes. Exact accuracy and consistency therefore answer different questions, and their percentages need not agree.