RESEARCH · LEARNING

Learning Relational Action Rules Without a Model

Inducing first-order rules from demonstrations, and a causal ablation of the learned competence

Stefan Ragland

Dominion Labs Research & Development

Dominion Labs

research@dmnlabs.org

November 12, 2024

Abstract

We study whether a symbolic learner can acquire general relational action rules purely from labeled demonstrations — before/action/after state triples — with no language model consulted at any point, and whether the resulting competence is *causally* carried by the learned rules. From a small demonstration set the learner induces two first-order rules with variables: a static derivation rule and a state-transition rule with an explicit deletion effect. Learning is verified model-free by an enforced policy that blocks and counts any model call (measured: zero throughout). On a frozen held-out evaluation of 14 cases containing only unseen constants — positive derivations, missing-precondition negatives, role-reversal traps, irrelevant-fact distractors, and multi-step transition chains — the fully-equipped system solves 14/14. We then ablate the learned state across seven conditions, each in its own isolated clone. Removing the learned rules, or leaving them present but denying them applicable status, drops the score to 7/14 — exactly the cases that require no derivation — while a sham clone-and-restore control and a semantic-store deletion control both retain 14/14, and reinstating the deleted rules recovers 14/14. The competence is thus localized to the induced, validated rules and to nothing else.

1 Rule induction, and why ablation

Much current generalization relies on large pretrained models whose behavior is hard to attribute. We ask whether general, variable-carrying relational rules can be learned directly from a handful of examples, verifiably without any model in the loop — the classical goal of inductive logic programming [1,2,3]. A learned rule must generalize over *argument structure*, not surface co-occurrence: it must fire on entirely unseen con-

stants, ignore irrelevant facts, refuse when a precondition is missing, and refuse when relations hold but point the wrong way (role reversal). It must also express *deletion* effects, so that applying a transition rule changes the world state rather than merely accumulating conclusions [4].

> *General relational action rules can be induced from a handful of demonstrations without any language model, and the acquired competence is causally attributable to the learned rules: a controlled ablation removes the capability by deleting the rules (or revoking their applicable status) and restores it by reinstating them, while unrelated state and the cloning procedure itself are shown to be non-causal.*

A system can pass a benchmark for the wrong reason. Demonstrating that a capability is *present* is not the same as showing *what carries it*. A causal ablation — remove the hypothesized carrier and watch the capability vanish, restore it and watch it return, with code, runtime, inputs, and model-availability held fixed — converts a performance claim into a mechanistic one.

2 Method

Demonstrations as evidence. A teacher supplies labeled examples only: each is a before-state, an optional action, and an after-state, tagged positive or negative. The teacher asserts no rule. Negative examples hold all-but-one precondition and show the effect *not* occurring, forcing each precondition into the rule body rather than leaving it as an incidental fact.

Generalization by anti-unification. From the positive examples the learner computes a least-general generalization, replacing constants with typed variables shared consistently across argument positions [5], and keeps only the body literals discriminated by the negatives. This yields a first-order rule with add-effects and, where the after-state drops a fact, an explicit delete-effect.

Validation gate. An induced rule becomes *applicable* only after it is confirmed on separate validation demonstrations it never saw during induction. Rules that are present but not validated remain non-applicable.

Evaluation by application only. A frozen suite of unseen cases is answered purely by applying already-persisted applicable rules — never by inducing a new one. Two policies are asserted before any component loads: no learned model may be consulted (attempts are counted and asserted zero), and during evaluation no rule may be induced or re-validated. The frozen policy is what makes a *negative* ablation result meaningful: an ablated system cannot silently re-derive the capability it was stripped of.

Ablation design. The hypothesized carrier (the learned rules) is removed, degraded, or restored across seven conditions, each run against its own cloned copy in its own process so no cached state leaks. A sham con-

dition clones and restores without ablating (controlling for the cloning procedure); a semantic-store-deletion condition is a negative control for a component that should not matter.

3 The learned rules

Two rules were induced (relation symbols are abstract; roles are what matter). A static derivation rule concludes a relation when four conditions on its two arguments hold. A transition rule fires on an action and its preconditions, adds the resulting location fact, and — the delete effect — retracts the origin fact, so applying it moves the world forward:

Rule	Induced from	Form (schematic)
Static derivation	6 demos (2 pos, 4 neg)	$R_1(x,y) \wedge P(x) \wedge Q(y) \wedge R_2(x,y) \rightarrow S(x,y)$
State transition	5 demos	$\text{Move}(x,b,a) \wedge \text{Open}(a) \wedge \text{Path}(b,a) \rightarrow \text{At}(x,a) \oplus \text{At}(x,b)$

Table 1. The two induced rules, both confirmed on held-out validation demonstrations. $\oplus$ marks the delete-effect (the origin fact is retracted on application).

4 Results

On the frozen suite of 14 cases — all using unseen constants: positive derivations, missing-precondition negatives, an irrelevant-fact distractor, role-reversal traps, and transition-sequence cases — the fully-equipped system solves 14/14 with zero model calls.

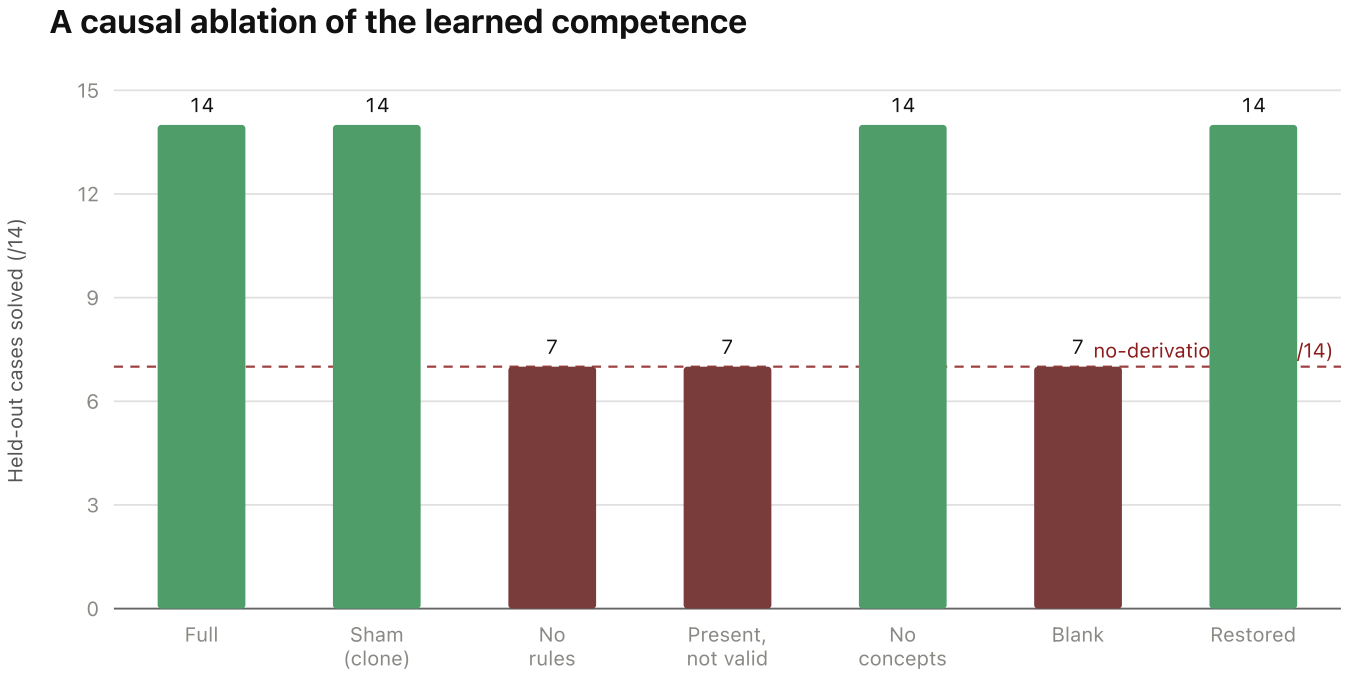

Figure 1. Held-out cases solved (out of 14) across seven conditions. Green retains full competence; red collapses to the 7/14 floor. Deleting the learned rules (No rules) or revoking their applicable status (Present, not valid) removes exactly the derivation-requiring cases; deleting the semantic concept store (No concepts) changes nothing; the sham clone-and-restore matches Full; and reinstating the rules (Restored) recovers 14/14.

Condition	Rules avail.	Applicable	Solved	Model calls
Full	2	2	14/14	0
Sham (clone + restore)	2	2	14/14	0
No learned rules	0	0	7/14	0
Present, not validated	2	0	7/14	0
No concepts (control)	2	2	14/14	0
Blank state	0	0	7/14	0
Restored	2	2	14/14	0

Table 2. The seven ablation conditions. Every condition ran with zero model attempts under an enforced model-free policy. “Present, not valid” isolates the validation gate itself: the rules exist but none is applicable, so the score is the same as deleting them.

The 7/14 floor is not noise: the seven cases that pass under every ablated condition are exactly the seven that require *no* derivation (missing-precondition refusals, role-reversal refusals, a blocked destination). The

seven that flip from pass to fail when the rules are removed or de-validated are exactly the seven requiring an actual derivation. This is the clean causal signature — removing the rules removes precisely the positive-derivation competence and nothing else.

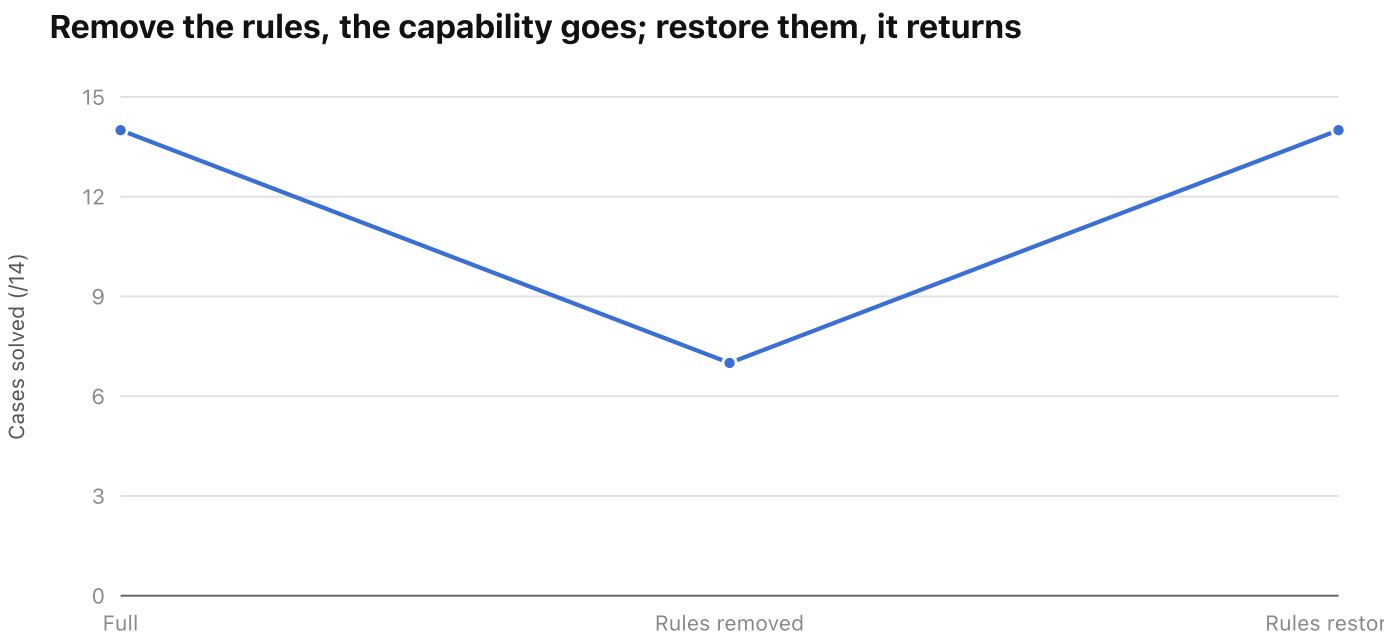

Figure 2. Reversibility: full competence (14/14), rules removed (7/14), rules reinstated (14/14). Because inputs, code, runtime, and model-availability are held fixed across the three, the learned rules are the difference that makes the difference.

5 Limitations and conclusion

The transition rule reported in every measured result was induced from five demonstrations, and its body omits an explicit source-location precondition; a later teaching set adds a sixth demonstration intended to force that precondition in, but the suite was not re-run with it, so we report the measured five-example rule. The relations are synthetic nonce symbols chosen to rule out lexical priors; the suites are fixed and modest in size. Within that scope the result is strong and, we think, unusually clean for a learning claim: two general relational rules are induced from a handful of demonstrations with no model in the loop, and a controlled ablation localizes the resulting competence to those rules — deleted, it vanishes; restored, it returns — while the concept store and the cloning procedure are shown to carry none of it.

References

[1] S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4), 1991.

[2] S. Muggleton, L. De Raedt. Inductive logic programming: theory and methods. *J. Logic Programming*, 19–20, 1994.

[3] S. Muggleton, D. Lin, A. Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic Datalog. *Machine Learning*, 100, 2015.

[4] R. E. Fikes, N. J. Nilsson. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3–4), 1971.

[5] G. D. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5, 1970.