ALDER: DISCOVERING THE LAWS OF A WORLD BY ACTING IN IT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable world models should not only predict future states but express how actions change the world in an explicit, transparent and testable form, such as equations. Yet methods that rely on a fixed set of trajectories cannot distinguish equally good competing hypotheses, while searches over a fix set of predefined candidates cannot discover equations outside the initial hypothesis space. We introduce ALDER (Action-guided Law Discovery, Evaluation, and Revision), a method that actively proposes novel experiments to test and revise models. Specifically, ALDER proposes parametric equations; a numerical optimizer fits their coefficients; an independent verifier tests these candidates on held-out data. To distinguish between competing valid hypotheses, a cost- and safety-aware selector queries interventions, in the form of novel experiments. The resulting counterexamples update the evidence ledger and guide the next structural revision, while incompatible laws are discarded. Across an in-house benchmark, ODE equation discovery tasks, and robotic experiments, ALDER discovers laws beyond its initial formula set, repairs failed model proposals, distinguishes fixed candidate models with fewer interactions, and improves out-of-distribution prediction. Furthermore, given a current state and a target, ALDER selects control actions by solving the inverse problem defined by its validated world model. Together, these results show that explicit equation-based world models can be tested and revised through interaction, then naturally used to guide goal-directed control.

1 INTRODUCTION

Autonomous agents need world models that can predict beyond their initial learning support. A world model must not only fit observed transitions but anticipate how untried actions affect the environment. Explicit behavioral models such as ESBM represent action consequences as inspectable and reusable structured knowledge (Shindo et al., 2026). For physical and robotic tasks, models must also predict action-induced changes in continuous states to support planning and control (Hafner et al., 2019a;b). Explicit equations offer an inspectable representation of dynamics (Brunton et al., 2016; Cranmer, 2023): they describe how states and actions determine predicted changes, allowing action outcomes to be simulated and predictions to be tested through new interactions. Learning reliable world models represented by equations from limited interaction is therefore an important problem connecting world modeling with model-based control.

Accurate predictions on the support set of trajectories do not guarantee an exact world model. Trajectories collected in limited scenarios typically cover only a small set of state–action combinations, on which structurally different equations may make nearly identical predictions. When object properties, action ranges, or contact conditions change, however, their predictions may diverge substantially. Fitting error on existing trajectories alone therefore cannot determine which model will predict reliably under new conditions; additional interactions are needed to test and distinguish these models by comparing their predictions with observed outcomes.

World model learning must go beyond fixed data and beyond predefined equations. Sparse-library methods construct equations from predefined terms but cannot introduce variables or operations absent from the library (Brunton et al., 2016; Champion et al., 2019). Symbolic regression can search for new expressions (Udrescu et al., 2020; Cranmer, 2023; Shojaee et al., 2025; Xia et al., 2026), but when existing data cannot distinguish candidate models, expanding the search alone cannot determine

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.

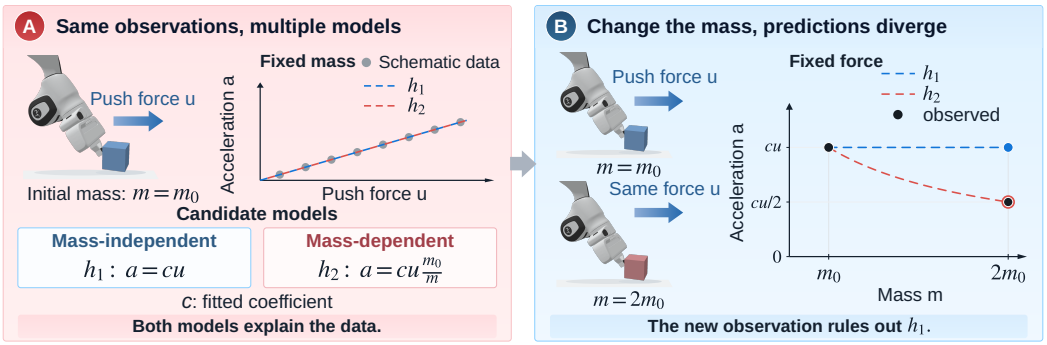


Figure 1: **Different world models can fit the same trajectories; ALDER selects experiments to distinguish them.** At mass m_0, both equations fit the observations (A). At $2m_0$ with the same force, their predicted accelerations differ. The observation contradicts the mass-independent prediction (B).

which model will predict correctly under new conditions. Active system identification can choose inputs to collect additional information, but formulations with prespecified model structures do not add missing equation terms (Wagenmaker & Jamieson, 2020). Addressing both problems therefore calls for experiments that distinguish candidate models and the ability to revise equation structures when fitted models fail to explain the observations.

Agents should set up targeted experiments that actively discard incorrect models. For each safe and feasible experiment, the system compares predictions from candidate world models and prioritizes experiments with larger predictive differences (Shyam et al., 2019). Observations can rule out models whose predictions substantially disagree with the measured outcomes. If no candidate explains the new observations, the existing equations may lack necessary variables or interaction terms, providing evidence for the next revision. Experimental outcomes thus not only add training data but help determine whether to select an existing model or revise its equation structure. Figure 1 contrasts fitting existing trajectories alone with this interactive world model learning process.

ALDER tests and revises equation-based world models. The system initially samples a set of trajectories in a shared record (*Evidence Ledger*). It then uses this evidence to propose or revise equation structures under type constraints (*Law Revision Agent*), and a numerical optimizer then fits their coefficients (*Coefficient Fitter*). When multiple candidate models explain the available data, an experiment selector chooses an experiment on which their predictions disagree, subject to cost and safety constraints (*Active Rollout Selector*, ARS). New observations are added to the ledger to guide subsequent model selection and structural revision. An independent verifier uses held-out data unavailable to the agent to decide whether to admit a candidate model (*Protected Admission Verifier*). For robotic control, an admitted equation, the nominal simulator, and a fixed low-level controller form a forward model that predicts the final state resulting from an action. Action selection is the corresponding inverse problem: given a target state, we numerically seek an action whose predicted final state is close to the target, then execute it through the controller. Rejected models undergo further revision, and no model is returned if none passes validation before the interaction budget is exhausted. Figure 2 shows this loop and the information flow between its components.

Our evaluation covers the full process from equation construction to robotic application. On custom benchmarks, ALDER constructs equations beyond its initial set and uses new observations to revise incorrect models in robotic tasks with tool-mediated pulling and non-convex object pushing. Public ODE datasets further test its ability to model different dynamical systems. With candidate models and budgets fixed, the ARS module reduces the mean number of interactions needed to distinguish them. In RoboCasa drawer and door tasks (Nasiriany et al., 2026), admitted models reduce prediction error under distribution shift while maintaining control performance. When required equation terms, state variables, or observations are missing, the independent verifier rejects models unsupported by its validation data.

We use ALDER to study how active sampling of environment interactions improves equation-based world models and how these models can be used to support action. Our contributions are threefold. **(1) Experimental observations can drive equation structure revision, not just param-**

eter fitting. ALDER connects experiment selection with structural revision, using newly acquired observations to test existing models and iteratively construct better hypotheses. Our experiments show how this process repairs initial proposals and yields effective models beyond the initial candidate set. **(2) Active experiments selection based on competing hypotheses disagreement can efficiently improve world-model refinement.** We fix candidate models and interaction budgets to isolate the impact of active experiments selection, finding that disagreement-based selection reduces the mean identification cost. **(3) Learning equation-based models support testable prediction and naturally allows control.** In robotic tasks, our equation-based models reduce prediction error under distribution shift. The explicit equations also allow to perform action selection by solving the inverse problem, connecting model learning to goal-directed execution while maintaining control performance.

2 BACKGROUND AND RELATED WORK

Learning an equation requires choosing its structure and fitting its coefficients. SINDy selects a sparse set of dynamics terms from a predefined function library (Brunton et al., 2016), whereas symbolic regression searches expressions by combining variables and operators (Cranmer, 2023). LLM-SR couples language-model program proposals with numerical coefficient fitting (Shojaee et al., 2025), while SR-Scientist uses an agent to analyze data and revise executable formulas (Xia et al., 2026). ALDER retains this separation of structure generation and coefficient fitting, representing world models as executable equations subject to type constraints.

Active experiments can distinguish models that are difficult to separate from existing data alone. Active system identification selects inputs to reduce model uncertainty (Wagenmaker & Jamieson, 2020), APPS queries informative phase-space regions (Jiang et al., 2025), and LLM-ACES selects new initial conditions using disagreement between candidate model predictions, then uses the new trajectories in subsequent symbolic search (Abhyankar et al., 2026). ALDER applies this idea to environment interaction: it selects experiments subject to feasibility, cost, and safety constraints, uses new observations to revise equations, and independently validates the resulting models.

Trajectories from failed executions can inform revisions to executable knowledge. Kintsugi repairs an executable policy knowledge base through constrained edits and verification (Cao et al., 2026), while ESBM updates symbolic behavioral models from task rollouts, adaptive questions, and world-model probes (Shindo et al., 2026). These works primarily revise skills, predicates, and symbolic rules. ALDER instead revises equations describing continuous state changes: an agent edits the equation structure, a numerical optimizer fits coefficients, and an independent verifier decides whether to admit the resulting model.

3 ALDER: ACTION-GUIDED LAW DISCOVERY, EVALUATION, AND REVISION

3.1 OVERVIEW AND EXECUTABLE LAWS

ALDER learns equation-based world models under a limited interaction budget. Its inputs are initial trajectories $\mathcal{D}_0$, a typed grammar $\mathcal{G}$ specifying available variables, operators, and their composition constraints, and a set of permitted environment experiments $\mathcal{Q}$. Each experiment $q \in \mathcal{Q}$ changes an initial state, object property, or action and returns a new trajectory $\tau(q)$. After k experiments, the data used for model learning become

$$\mathcal{D}_k = \mathcal{D}_0 \cup \{\tau(q_1), \ldots, \tau(q_k)\}. \tag{1}$$

Within the proposal and interaction budgets $\mathcal{B}$, the system constructs and tests candidate equations. It returns an independently validated equation structure h, fitted coefficients $\widehat{\xi}_h$, and a stated scope of applicability; if no candidate passes validation when the process terminates, it refuses to return a model:

$$\text{ALDER}(\mathcal{D}_0, \mathcal{G}, \mathcal{Q}, \mathcal{B}) \longrightarrow (h, \widehat{\xi}_h) \quad \text{or} \quad \perp. \tag{2}$$

Here, $\perp$ denotes refusal to return a model.

Expression structure and numerical coefficients together define the model's dynamics equations. Structure h is an executable expression tree specifying how variables and operators are combined; it belongs to the set $\mathcal{H}(\mathcal{G})$ permitted by the typed grammar $\mathcal{G}$. The agent generates or revises h, while

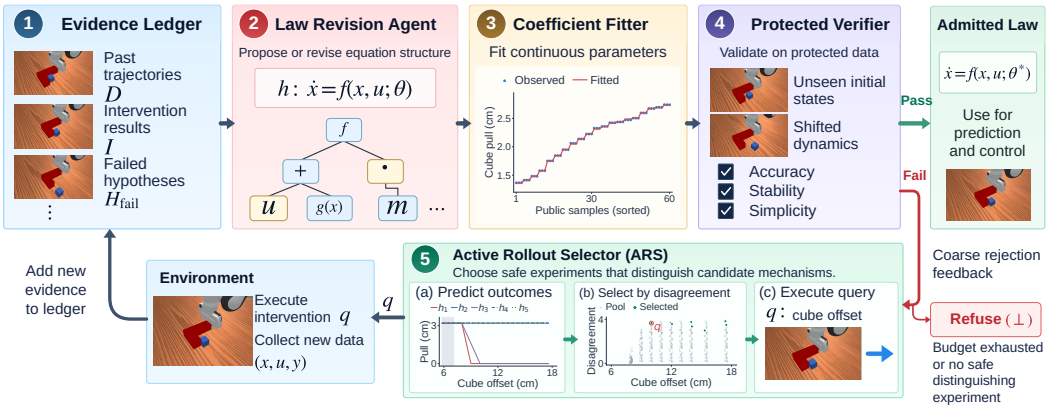


Figure 2: **ALDER uses new observations to revise equations.** The Law Revision Agent proposes an equation structure from public evidence, and the Coefficient Fitter estimates its continuous parameters. The Protected Admission Verifier checks fitted equations on held-out data; admitted equations can be used for prediction and action selection. After rejection, ARS may select a feasible experiment where candidate equations predict different outcomes; its observation guides the next revision.

the Coefficient Fitter estimates ξ_h within prescribed bounds from the accumulated trajectories. Given input e_t containing task-relevant states, geometry, and actions, executing the equation predicts the task response y_t:

$$\widehat{y}_t = \phi_h(e_t; \xi_h), \qquad h \in \mathcal{H}(\mathcal{G}). \tag{3}$$

Here, ϕ_h evaluates the equation specified by h. Depending on the task, the predicted quantity may be acceleration, force, or a score indicating whether an event occurs. Fixed dynamics calculations and numerical integration convert predicted derivatives or forces into future states. This representation supports object interactions, joint motion, and contact constraints. When dynamics admit a pairwise decomposition, the same equation can be shared across object pairs. The equation grammar and validation scope are detailed in Appendices A.1.3 and A.1.1.

ALDER iteratively revises candidate equations and uses independent validation to decide whether to admit them. The *Evidence Ledger* stores initial trajectories, experimental observations, and previous failures. The *Law Revision Agent* uses these records to propose or modify equation structures, and the *Coefficient Fitter* fits their coefficients. The *Protected Admission Verifier* evaluates candidate models on held-out data unavailable to the agent. Following rejection, the *Active Rollout Selector* (ARS) may select a feasible experiment on which candidate models predict different outcomes. The resulting observation is added to the ledger for the next revision. The system returns a model that passes validation; if no candidate passes within the budget, it refuses to return a model.

3.2 DISCOVERING LAWS BY ACTING

ALDER revises the equations structures using observed trajectories and failure records. At round k, the Evidence Ledger provides variable meanings, current trajectories, fit errors and residual analyses, executed experiments, and permitted rejection categories. In an isolated sandbox, the Law Revision Agent reads these records and the typed grammar $\mathcal{G}$ to produce a candidate structure h_k as a JSON expression tree, together with parameter names and bounds, an evidence-based rationale, and an intended scope. Only candidates passing grammar, type, expression-size, and executability checks proceed to coefficient fitting and validation by the subsequent components. The agent proposes structures; it neither fits coefficients nor decides whether a model is admitted.

Once the structure is fixed, the Coefficient Fitter estimates only numerical coefficients. We extract N_k input–target pairs (e_n, y_n) from the current trajectories $\mathcal{D}_k$ and minimize prediction loss over the permitted coefficient domain Ξ_{h_k}:

$$\widehat{\xi}_{h_k} = \arg\min_{\xi \in \Xi_{h_k}} \frac{1}{N_k} \sum_{n=1}^{N_k} \ell(\phi_{h_k}(e_n; \xi), y_n)\,. \tag{4}$$

We use squared error for continuous targets and class-balanced logistic loss for events. Squared-error models linear in the coefficients use ridge regression; nonlinear parameters use bounded reparameterization and gradient optimization (Appendix A.1.2). Fitting updates only coefficients; missing variables, operators, or compositions require the agent to revise the equation structure.

The Active Rollout Selector (ARS) uses prediction disagreement to choose experiments while accounting for safety and cost. At round k, the candidate set contains equation structures, fitted coefficients, and weights:

$$\mathcal{C}_k = \left\{ (h_m, \widehat{\xi}_m, w_m) \right\}_{m=1}^{M_k}, \qquad \sum_{m=1}^{M_k} w_m = 1. \tag{5}$$

Candidates come from current or earlier proposals, grammar-permitted alternatives, or a supplied set. All are refitted on the same $\mathcal{D}_k$ and weighted by public validation likelihood, error, or BIC; weights are uniform when no calibrated score is available. The set need not contain the true equation. Neither protected data nor actual outcomes of unqueried experiments inform its construction or experiment selection. For each feasible experiment q, candidates predict trajectories $\widehat{\tau}_{h_m}(q)$, and ARS selects

$$q^\star = \arg \max_{q \in \mathcal{Q}: S(q)=1} \left[\sum_{a<b} \widetilde{w}_{ab} d(\widehat{\tau}_{h_a}(q), \widehat{\tau}_{h_b}(q)) + \lambda_D D(q, \mathcal{D}_k) - \lambda_c c(q) \right], \tag{6}$$

where d measures prediction disagreement, $\widetilde{w}_{ab}$ emphasizes plausible model pairs, D rewards experimental inputs unlike those already observed, c measures interaction cost, and $S(q) = 1$ denotes satisfaction of safety constraints. A rejected proposal can also be compared directly with alternative models to guide further experiments. Prediction disagreement serves as a proxy for information gain, not an exact mutual-information calculation.

The novel experiments allow to discriminate candidate models and inform equation revision. The environment executes the selected experiment $q^\star$ and adds its trajectory to the dataset through Equation 1. An observation that supports some candidates and contradicts others helps distinguish models; if no candidate explains it, the equation structures may need revision. The Evidence Ledger records failed predictions, the corresponding state–action conditions, and public residual diagnostics. Using this evidence, the Law Revision Agent may add or remove variables, change operators, or revise compositions and piecewise branches within $\mathcal{G}$. After each edit, the Coefficient Fitter refits coefficients from scratch, and the independent verifier decides whether to admit the candidate model.

3.3 Independent Admission and Downstream Use

Candidate models are admitted only after passing independent validation. The Protected Admission Verifier uses protected data unavailable to the agent to check prediction error, improvement over a simple reference, expression complexity, numerical stability, finite outputs, and performance under task-specific distribution shifts. Separate public splits support coefficient fitting and diagnostics; final test data do not participate in model construction, experiment selection, or admission. After rejection, the agent receives only public diagnostics and a coarse failure category, never protected data or scores, the target equation, or final test outcomes. A candidate passing all preset criteria is returned as $(h, \widehat{\xi}_h)$. Otherwise, revision continues until a candidate passes, no distinguishing safe experiment remains, or the proposal or interaction budget is exhausted; the latter two cases return $\perp$.

A validated world model supports control as an inverse problem. Given state X_0, the learned joint-force equation, nominal simulator, and fixed low-level controller jointly predict the joint position $\widehat{q}_T(u)$ after executing action u for T steps. For target position g, control seeks an action satisfying

$$|\widehat{q}_T(u^\star) - g| \le \varepsilon. \tag{7}$$

With the equation and coefficients fixed, numerical root finding searches bounded actions over preset horizons. Within the model-evaluation budget, we return an action meeting target tolerance ε and predicted safety constraints; otherwise, we return the closest-to-target evaluated action satisfying the safety constraints. Without a validated model or an action satisfying these safety constraints, execution is refused. Control success is measured from actual execution.

Table 1: **Interactive revision improves functional recovery; fitted equations predict ODEBase trajectories under shift.** A: NovelLaw reports functional recovery (%) and median shifted NMSE over 60 cases; robot shifted NMSE is mean ± SD over five episode-disjoint partitions. Rejected runs are included. B: median trajectory NMSE and mean symbolic accuracy on 59 ODEBase systems.

A. Controlled discovery

	NovelLaw (60 cases)		PullCubeTool (5 runs)		PushT (5 runs)	
Method	Functional recovery (%, ↑)	Shift NMSE (↓)	Admitted (↑)	Shift NMSE (↓)	Admitted (↑)	Shift NMSE (↓)
One-shot agent	43.3	6.54×10^{-2}	0/5	4.816 ± 1.973	3/5	0.842 ± 0.644
Revision-only agent	75.0	1.50×10^{-2}	0/5	1.296 ± 0.024	3/5	1.292 ± 1.638
Random-query agent	**85.0**	9.15×10^{-3}	**5/5**	0.194 ± 0.138	**4/5**	0.673 ± 0.473
ALDER	**85.0**	$\mathbf{5.58\times10^{-3}}$	**5/5**	$\mathbf{0.161 \pm 0.052}$	**4/5**	$\mathbf{0.420 \pm 0.185}$

B. ODEBase: published baselines

Method	Reconstruction NMSE (↓)	Generalization NMSE (↓)	Future-time NMSE (↓)	Symbolic accuracy (%, ↑)
SINDy (Brunton et al., 2016)	1.06e-03	1.12e+00	3.47e-01	5.9
PySR (Cranmer, 2023)	1.37e-03	1.08e+00	1.35e+00	5.9
QBC (Abhyankar et al., 2026)	4.97e-09	7.47e-06	6.38e-06	14.1
BO (Abhyankar et al., 2026)	8.55e-10	4.61e-10	5.43e-08	49.3
APPS-ODE (Jiang et al., 2025)	5.56e-01	9.12e-01	1.01e+00	15.2
LLM-ACES (Abhyankar et al., 2026)	3.70e-15	4.18e-15	3.44e-13	52.4
ALDER	**2.32e-15**	**2.28e-16**	**3.28e-29**	**90.7**

4 EXPERIMENTS

Reliable world model learning must address accurate model construction, interaction efficiency, reliable validation and can allow control utility. We evaluate ALDER using four research questions. **RQ1:** Can ALDER construct equations beyond its initial set and use failure evidence to revise candidate models? **RQ2:** With candidate models and interaction budgets fixed, can ARS identify the target model with fewer interactions? **RQ3:** Do validated models improve prediction under distribution shift and support control in RoboCasa? **RQ4:** When equation terms, state variables, or observations are insufficient, does the system refuse models unsupported by the available evidence?

Experimental setup. Except for oracle references with privileged information, methods access only public trajectories and permitted intervention outcomes, not hidden environment ground truth, protected admission data, or final test data. Controlled comparisons use the same splits and match observations, query pools, coefficient fitters, and interaction budgets according to the comparison's purpose. Invalid proposals and refusals remain in the denominators for model recovery and control success. Full protocols are provided in Appendix A.

4.1 ALDER CONSTRUCTS AND REPAIRS LAWS BEYOND THE INITIAL SET (RQ1)

To evaluate how well models discover and update dynamical equations, we test them on both custom interaction tasks and public benchmark systems. Our *NovelLaw* benchmark evaluates out-of-set equation construction, while custom *PullCubeTool* and *PushT* protocols in ManiSkill (Tao et al., 2025) test whether replayed actions over varied poses drive revision of contact-dependent dynamics. Finally, the ODEBase benchmark (Lüders et al., 2022) provides 59 dynamical systems to assess construction at scale, with full setups and additional benchmarks detailed in Appendix A.

We evaluate baseline variants to isolate the individual contributions of structural revision and interactive querying. One-shot acts as a single-proposal reference, revision-only tests iterative editing without new data, and random-query shares ALDER's interaction budget but selects queries uniformly at random. All variants use identical grammars, coefficient fitters, and admission criteria. Table 5 adds fixed-library, SINDy, and MLP baselines for comparison against fixed structures, sparse regression, and neural networks. On ODEBase, we benchmark against published sparse, symbolic, and active discovery baselines (Table 1B; Appendix A.6.1).

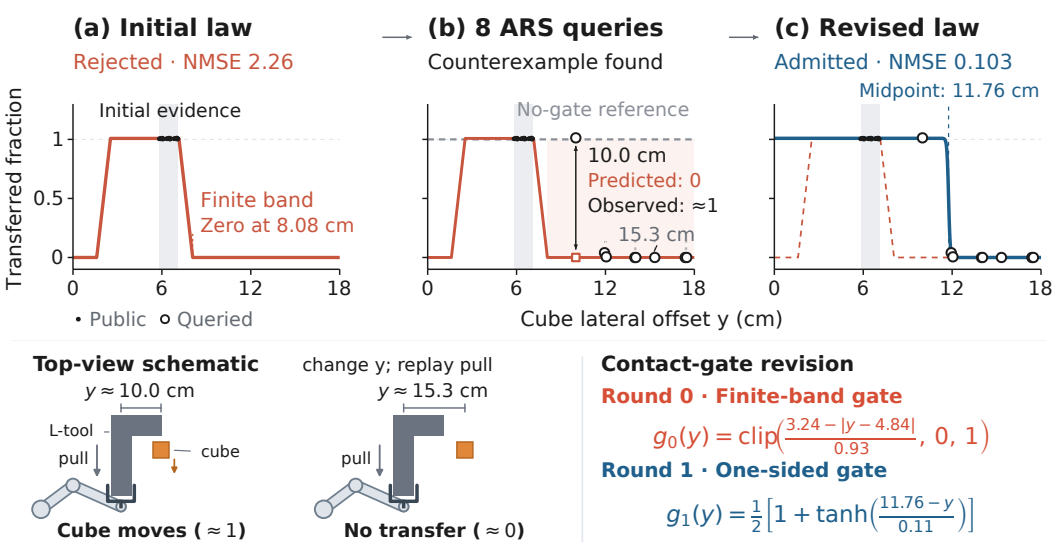


Figure 3: **An ARS query reveals a failed prediction and guides revision of the PullCubeTool contact gate.** (a) The initial equation fits demonstrations but fails the admission checks. (b) At 10 cm, the observed transfer fraction is near one, but the equation predicts zero. (c) The equation with a revised one-sided gate is admitted. Filled and hollow points mark public and queried observations; some query points overlap. The schematics show the same pull at two cube positions; the formulas describe only the contact gates. NMSE is measured on the protected admission set.

Interaction evidence helps ALDER revise initial equations and improve prediction under distribution shift. On NovelLaw, ALDER retains every initial success and increases recovery from 26/60 to 51/60, matching random-query's recovery with lower shifted NMSE. On PullCubeTool, all ALDER runs pass predictive validation, whereas one-shot and revision-only never pass. Across both robot tasks, ALDER matches random-query's admission counts but has lower mean shifted error (Table 1A). Figure 3 shows how a new observation contradicts the initial prediction and prompts revision of the contact gate from a finite band to a one-sided structure. On ODEBase, 49/59 systems pass the joint functional test across reconstruction, unseen initial conditions, and future-time prediction. Table 1B additionally reports prediction errors and symbolic accuracy, with published results from Abhyankar et al. (2026) as references. These comparisons are not matched for model backbone or interaction budget. Appendix A.6.1 identifies the sources of the reference results and describes our independently implemented procedure for evaluating symbolic accuracy.

4.2 ARS IDENTIFIES MECHANISMS FASTER THAN RANDOM QUERIES (RQ2)

Our *NovelLaw* benchmark evaluates out-of-set equation construction, while custom *PullCubeTool* and *PushT* protocols in ManiSkill (Tao et al., 2025) test whether replayed actions over varied poses drive revision of contact-dependent dynamics. Finally, the ODEBase benchmark (Lüders et al., 2022) provides 59 dynamical systems to assess construction at scale, with full setups and additional benchmarks detailed in Appendix A.

Active queries driven by mechanism disagreement consistently minimize identification cost across all tested baselines. ALDER identifies every target with a median identification cost of one, whereas random selection has a median cost of three and leaves some targets unresolved (Figure 4). Their mean penalized identification costs are 1.16 and 3.53, respectively. Geometric coverage also identifies every target and, like parameter uncertainty, has a median cost of one. Their mean costs, however, are 1.91 and 1.35, both above ALDER. Paired statistics support these mean-cost advantages over both strong baselines, showing that prediction disagreement can further improve identification efficiency when the candidate models are given in advance (Appendix A.3).

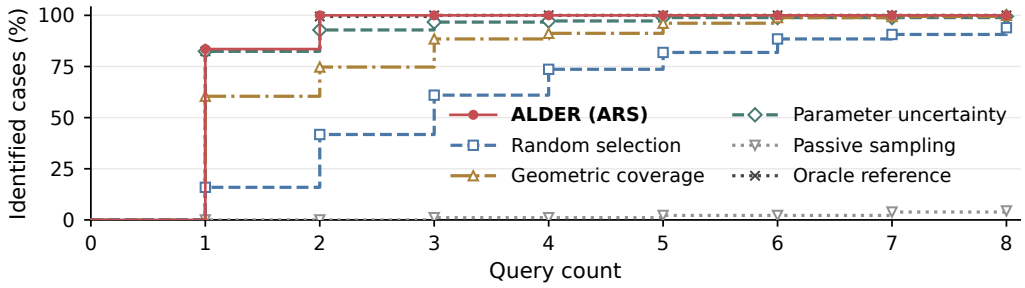


Figure 4: **In complete eight-query records, ARS identifies the target model earlier than random selection.** Curves show the cumulative fraction reaching sustained correct identification among fixed candidate models, computed retrospectively. Unresolved cases remain in the denominator.

4.3 Explicit Laws Improve Shifted Prediction and Preserve Control (RQ3)

We test world models for prediction and control in RoboCasa's `OpenDrawer` and `OpenDoor` tasks (Nasiriany et al., 2026; Zhu et al., 2020). A fixed controller first grasps the handle; the validated world model then predicts pull outcomes, and numerical inversion selects pull amplitude and duration for a target joint position. We vary joint damping, Coulomb friction, spring stiffness, and their composition in MuJoCo (Todorov et al., 2012) to test performance under changing dynamics. Each task combines four dynamics families with ten random seeds, yielding 40 simulated worlds. Models use simulator-provided state and force observations.

Table 2: **Explicit dynamics discovered by ALDER improve shifted prediction while preserving downstream control performance in RoboCasa.** Normalized joint-position RMSE and action-selection success rates are reported as mean ± standard deviation across 10 random seeds, each evaluated over four mechanism families (640 trials per cell) in the *OpenDrawer* and *OpenDoor* tasks. Bold values denote the best non-oracle performance. The parameter and outcome oracles serve as evaluation references using ground-truth parameters and actual execution outcomes, respectively.

	OpenDrawer			OpenDoor		
Model	OOD prediction RMSE (↓)	Control MAE (↓)	Success (%, ↑)	OOD prediction RMSE (↓)	Control MAE (↓)	Success (%, ↑)
Nominal	6.8 ± 0.7	5.04 ± 0.54	83.6 ± 2.2	7.2 ± 0.8	4.26 ± 0.37	94.4 ± 2.0
Passive law	7.5 ± 1.0	5.78 ± 0.76	80.3 ± 3.6	7.4 ± 1.5	3.80 ± 0.49	89.7 ± 3.1
Matched MLP	3.7 ± 0.8	1.98 ± 0.35	96.9 ± 2.4	3.1 ± 0.7	1.81 ± 0.32	99.5 ± 1.1
ALDER	$\mathbf{0.9 \pm 0.1}$	$\mathbf{0.74 \pm 0.10}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{0.9 \pm 0.1}$	$\mathbf{0.49 \pm 0.06}$	$\mathbf{100.0 \pm 0.0}$

We use a common numerical inverse solver to test the effects of interactive revision and model representation. The nominal model does not adapt to dynamics changes and provides an initial reference. The passive law uses only initial trajectories, testing the joint benefit of new interactions and subsequent revision. The matched MLP and ALDER share training data, action ranges, and solver budgets to compare neural and equation-based models. We report shifted prediction error, actual endpoint-to-target error, and control success. Success requires safe actual execution and normalized target-position error at most 0.01. Full settings and supplementary results appear in Appendix A.4.

Across both tasks, validated equation-based models reduce prediction and control errors while maintaining high target-reaching success. With the same training data and numerical solver, ALDER predicts more accurately under distribution shift than the matched MLP, and its selected actions yield endpoints closer to the targets. Both achieve high control success, while ALDER meets the success criteria on all tested targets (Table 2). Relative to the passive law trained only on initial trajectories, ALDER improves prediction accuracy, control accuracy, and success, supporting the joint benefit of new interaction evidence and subsequent equation revision.

Table 3: **Admission checks reject models with missing equation terms or state variables; degraded force observations reduce prediction coverage.** A: AUROC and gated-residual NMSE are evaluated under distribution shift and do not determine admission. B: Drawer and Door are pooled; coverage is the fraction of worlds with admitted equations. RMSE uses those worlds only, whereas control success covers all targets, counting refusals as failures. Values with $\pm$ are mean $\pm$ SD.

A. Missing structure or state

Regime	Admitted models	Localization AUROC (↑)	Gated NMSE ($\times 10^{-3}$, ↓)
Complete	100.0%	–	0.042 ± 0.029
Missing smooth	0.0%	0.494 ± 0.019	14.5 ± 3.0
Missing local	0.0%	0.969 ± 0.007	18.5 ± 2.9
Hidden local	0.0%	0.997 ± 0.001	168 ± 114

B. Robot observation channel

Channel	Pred. coverage	RMSE ($\times 10^{-3}$, ↓)	Control success (↑)
Force reference	100.0%	0.9 ± 0.7	100.0%
Force noise 1%	80.0%	1.0 ± 0.7	79.9%
Force noise 5%	58.8%	1.0 ± 0.7	58.8%
Force noise 10%	42.5%	1.2 ± 0.7	42.5%
Estimated force	0.0%	–	0.0%

4.4 Verification Rejects Laws Unsupported by Available Evidence (RQ4)

We separately restrict equation structure and available observations to test rejection of unsupported world models. With the library fixed, we compare complete models, missing global smooth terms, missing local contact terms, and hidden key state over 20 seeds per regime, using an auxiliary residual diagnostic to locate states where errors concentrate. In RoboCasa's Drawer and Door tasks, we add noise to force observations or estimate force from position, velocity, and known control actions to assess how limited observations affect admission, prediction coverage, and control success. Control uses the same numerical inverse solver and success criteria as RQ3. Full settings appear in Appendices A.5.1 and A.5.2.

Under the tested conditions, ALDER rejects incomplete models and limits admission as observation quality degrades. All complete models pass verification. For missing local terms or key state, all models are refused and auxiliary localization is effective. With a missing global smooth term, all models are still refused despite near-chance localization under distribution shift (Table 3A). In the robot tasks, higher force-observation noise reduces prediction coverage; models fitted to the current motion-based force estimates do not pass validation (Table 3B). Prediction error remains low among admitted models, whereas control success is computed over all targets, with refusals and unsuccessful executions both counted as failures.

5 Limitations and Future Work

Future work will expand ALDER's model expressiveness and robot applications while improving learning and verification reliability. The current method uses a given state representation and mathematical grammar, with experiments mainly based on short simulated interactions and state and force observations. Extensions can expand equation operators and intervention interfaces, incorporate visual perception and contact trajectory information, and evaluate perception error, interaction cost, and safety on physical robots. Candidate models may share prediction errors, and reusing validation data may lead to overfitting. Increasing candidate diversity, limiting verification feedback, and testing on independent data offer directions for improving reliability and studying theoretical guarantees.

6 Conclusion

We introduce ALDER to actively test and revise equation-based world models through environment interaction. Prediction disagreement among candidate models guides experiment selection, interaction counterevidence drives revision of equation structure, and independent verification determines whether models are admitted for prediction and control. Experiments show that ALDER constructs equations beyond its initial set, identifies target models earlier among given candidates, improves shifted prediction while preserving robot control performance, and refuses unsupported models. Environment interaction thus provides not only training data but also a tool for identifying, falsifying, and revising world models.

AI USE STATEMENT

In RQ1, a Codex agent proposes typed formula programs inside isolated repositories using only protocol-defined public evidence. RQ2 fixes candidate formulas and compares numerical query selectors without generative formula proposals. In RQ3, GPT-5.6 proposes and revises joint dynamics for RoboCasa prediction and action selection; open-weight models replace the proposal backend in the corresponding ablation. RQ4 tests numerical admission under missing structure, hidden state, and degraded observations rather than using an LLM to judge candidate validity. Across these experiments, proposal models cannot access protected targets or determine evaluation scores; controller-owned code fits coefficients and computes admission decisions and metrics. Generative AI assisted method implementation. AI tools were used to polish the manuscript.

ETHICS STATEMENT

The reported study uses generated data, public demonstrations, and robot simulation without human subjects or physical intervention. No personal data or hardware experiment is included. Active experiments on physical robots could create safety risks; future deployment must enforce independent workspace, velocity, force, and collision limits together with human-authorized emergency stops. Model/API and interaction costs will be disclosed with the artifact.

REPRODUCIBILITY STATEMENT

The released artifact records every ownership boundary and final result needed to reconstruct the evaluation. Section 3 specifies proposal, fitting, acquisition, revision, and admission. Appendix A records frozen grammars, budgets, splits, and statistical protocols. Per-seed outputs, raw agent events, candidate programs, query indices, protected verifier decisions, and figure/table source data are retained. Automated checks cover data-split integrity and information leakage; 178 core implementation tests, 23 completion-package tests, and 28 RoboCasa-specific tests pass. The final RoboCasa statistics were independently recomputed and confirmed against the archived results. Every reported value can be traced to the corresponding experiment outputs.

REFERENCES

Nikhil Abhyankar, Sha Li, Sanchit Kabra, Naren Ramakrishnan, Yulia Gel, and Chandan K. Reddy. LLM-ACES: Closed-loop discovery of dynamical systems with LLM-guided adaptive search. *arXiv preprint arXiv:2606.25039*, 2026. URL https://arxiv.org/abs/2606.25039.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Teng Cao, Yu Deng, Hikaru Shindo, Quentin Delfosse, Lanxi Wen, Suli Wang, Jannis Blüml, Christopher Tauchmann, and Kristian Kersting. Kintsugi: Learning policies by repairing executable knowledge bases. *arXiv preprint arXiv:2605.09487*, 2026. URL https://arxiv.org/abs/2605.09487.

Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116 (45):22445–22451, 2019. doi: 10.1073/pnas.1906995116.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*, 2023. URL https://arxiv.org/abs/2305.01582.

Stéphane d'Ascoli, Sören Becker, Philippe Schwaller, Alexander Mathis, and Niki Kilbertus. ODE-Former: Symbolic regression of dynamical systems with transformers. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5ed5c3c846f684a54975ad7a2525199f-Abstract-Conference.html.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. URL https://arxiv.org/abs/2004.07219.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Nan Jiang, Md Nasim, and Yexiang Xue. Active symbolic discovery of ordinary differential equations via phase portrait sketching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17626–17634, 2025. doi: 10.1609/aaai.v39i17.33938.

Christoph Lüders, Thomas Sturm, and Ovidiu Radulescu. ODEbase: A repository of ODE systems for systems biology. *Bioinformatics Advances*, 2(1):vbac027, 2022. doi: 10.1093/bioadv/vbac027. URL https://doi.org/10.1093/bioadv/vbac027.

Soroush Nasiriany, Sep Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05003fdb1e9562ab0c0a9719ea4de10-Abstract-Conference.html.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.049. URL https://www.roboticsproceedings.org/rss14/p49.html.

Hikaru Shindo, Yu Deng, Teng Cao, Quentin Delfosse, Christopher Tauchmann, Jannis Blüml, Gopika Sudhakaran, and Kristian Kersting. Learning explicit behavioral models with adaptive questions and world-model probes. *arXiv preprint arXiv:2606.07127*, 2026. URL https://arxiv.org/abs/2606.07127.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 16054–16085, 2025.

Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5779–5788. PMLR, 2019.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU parallelized robot simulation and rendering for generalizable embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.021. URL https://www.roboticsproceedings.org/rss21/p021.html.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/33a854e247155d590883b93bca53848a-Abstract.html.

Andrew Wagenmaker and Kevin Jamieson. Active learning for identification of linear dynamical systems. In *Proceedings of the 33rd Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3487–3582. PMLR, 2020.

Shijie Xia, Yuhan Sun, and Pengfei Liu. SR-Scientist: Scientific equation discovery with agentic AI. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KBN6oUx5uL.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020. URL https://arxiv.org/abs/2009.12293.

A Implementation and Evaluation Details

This appendix follows the implementation and four main evaluations, then presents additional validation and reproducibility details. Appendix A.1 specifies the equation-based model implementation. Appendices A.2–A.5 cover construction, identification, control, and verification. Appendix A.6 reports additional systems and proposal backbones, and Appendix A.7 reports statistical protocols, model settings, and resource use.

A.1 ALDER Separates Proposal, Fitting, and Verification

A.1.1 Meaning and Scope of an Executable Equation

An executable equation combines a structure, fitted coefficients, and a stated prediction scope. The fitted pair $(h, \widehat{\xi}_h)$ defines this equation over its stated inputs and outputs. The typed grammar $\mathcal{G}$ specifies the available variables and operators; $h \in \mathcal{H}(\mathcal{G})$ selects a permitted equation structure, and $\widehat{\xi}_h$ contains its fitted numerical coefficients. For task input e_t, the program $\phi_h(e_t; \widehat{\xi}_h)$ predicts the task-defined response y_t. Admission means that this fitted program passes the predeclared checks on harness-held validation data, including applicable shifted regimes. These checks support prediction in the tested scope; they do not establish that the program uniquely describes the underlying physical dynamics. Recovery against a known reference equation is assessed separately, after discovery is complete.

A.1.2 Procedure and Coefficient Fitting

The experiment harness separates proposal, fitting, experiment selection, and protected verification throughout the discovery loop. Here, the harness is the software that runs the discovery procedure, distinct from the low-level robot controller used for execution. Algorithm 1 gives the complete procedure. The grammar, split roles, proposal and query budgets, admission thresholds, and final evaluation seeds are frozen before discovery begins.

Algorithm 1 ALDER's Verified Law Discovery Loop

Input: Public evidence $\mathcal{D}_0$, grammar $\mathcal{G}$, feasible experiments $\mathcal{Q}$, and budgets $\mathcal{B}$.
Input: Protected admission and shifted validation data $\mathcal{V}$.
Output: A validated model $(h, \widehat{\xi}_h)$, or refusal $\perp$.

Initialize the *Evidence Ledger* $\mathcal{E}_0$ from $\mathcal{D}_0$; set $k \leftarrow 0$.
Create an isolated repository containing only public evidence, grammar, instructions, proposal schema, and permitted failure records.
while proposal budget remains **do**
  $h_k \leftarrow$ *Law Revision Agent*$(\mathcal{E}_k, \mathcal{G})$; charge the proposal budget.
  if schema, arity, variable, bound, size, or execution checks fail **then**
    Record the public validation failure in $\mathcal{E}_k$; **continue**.
  end if
  Compile h_k; fit $\widehat{\xi}_{h_k}$ with the *Coefficient Fitter* on $\mathcal{D}_k$ only.
  Record public fit diagnostics in $\mathcal{E}_k$.
  $(a_k, r_k) \leftarrow$ *Protected Admission Verifier* $(h_k, \widehat{\xi}_{h_k}; \mathcal{V})$.
  if $a_k =$ accept **then**
    Freeze $(h_k, \widehat{\xi}_{h_k})$ and $\mathcal{E}_k$; **return** $(h_k, \widehat{\xi}_{h_k})$.
  end if
  Add only the permitted rejection category r_k to $\mathcal{E}_k$.
  if proposal or query budget is exhausted **then break**.
  Build and refit candidate models $\mathcal{C}_k$ using public evidence only.
  $q^\star \leftarrow$ *Active Rollout Selector* $(\mathcal{C}_k, \mathcal{D}_k, \mathcal{Q})$ within the remaining budget, without accessing unrevealed outcomes.
  if no safe distinguishing experiment is available **then break**.
  Commit $q^\star$, obtain its outcome $\tau(q^\star)$, and charge the query budget.
  $\mathcal{D}_{k+1} \leftarrow \mathcal{D}_k \cup \{\tau(q^\star)\}$.
  $\mathcal{E}_{k+1} \leftarrow \mathrm{Update}(\mathcal{E}_k, \tau(q^\star),$ public failure diagnostics$)$; $k \leftarrow k+1$.
end while
Freeze the unresolved ledger; **return** $\perp$.

Table 4: **Typed grammars expose building blocks, not complete target equations.** In RQ1, the variables and primitive operators are available, but complete target expressions and harness-held reference bases are absent from the proposal sandbox.

Track	Variables	Additional operators	Output / limit
NovelLaw	distance, radial speed, masses, radii, radial action	protected division, literal power, exponential, $\tanh$	acceleration; 31 nodes
PullCubeTool	tool-frame cube pose, tool/cube velocities, relative axis, commanded pull	protected division, power, exponential, trigonometric functions, absolute value, maximum	pull-axis displacement; 95 nodes
PushT	T-frame pusher pose, no-object TCP displacement, swept-contact indicator	protected division, power, exponential, trigonometric functions, absolute value, maximum	angular displacement; 95 nodes
Door	current and initial hinge/latch angles	literal power, sine, cosine	displacement; 127 nodes
Peg	hole-frame x, y, z and hole radius	literal power, absolute value, pairwise maximum	signed score; 63 nodes

Recovery against reference equations is evaluated outside the discovery loop. After the program or unresolved ledger is frozen, recovery is evaluated once on evaluator-only points. These points never enter proposal, coefficient fitting, experiment selection, or admission.

Coefficient fitting optimizes numerical parameters without changing program structure. For squared-error objectives linear in the coefficients, the fitter uses a ridge-regularized linear solve. For nonlinear parameters, each bounded coefficient $\xi_r \in [a_r, b_r]$ is represented as

$$\xi_r = a_r + (b_r - a_r)\,\sigma(\rho_r), \tag{8}$$

where σ is the logistic sigmoid. The unconstrained ρ_r is updated by backpropagation with multiple restarts and gradient clipping. This reparameterization enforces the proposal's declared coefficient bounds during optimization.

A.1.3 Typed Formula Grammar

Every typed grammar combines shared safe arithmetic with a benchmark-specific operator subset. All grammars contain variables, finite constants, named bounded parameters, addition, subtraction, and multiplication. Protected division floors denominator magnitude at 10^{-6}, exponentials are clipped, and powers receive a positive base floor. The harness independently validates, compiles, and refits every submitted program.

Pairwise decomposition is one instance of the equation-based model representation. Inputs to Equation 3 may include relative object states, joint position and velocity, geometric margins, or applied actions. When a multi-object system admits a pairwise decomposition, the same local program is shared across object pairs:

$$\widehat{\dot{x}}_t^{\,i} = f_{\text{self}}(x_t^i, u_t) + \sum_{j \neq i} \phi_h(e_{ij,t}; \xi_h). \tag{9}$$

Here, f_{self} describes the object's self dynamics and $e_{ij,t}$ contains pairwise features. This decomposition is not required for articulated systems or contact constraints represented directly by Equation 3.

A.1.4 Sandbox and Leakage Controls

Each NovelLaw proposal runs in a sandbox that excludes protected evidence. Every case uses a separate nested Git repository. The proposal agent sees a Bubblewrap namespace in which the parent project is absent, inherited environment variables are cleared, network use is disallowed by task contract, and temporary authentication material is removed after session startup. The target expression, generator seed, active-pool outcomes, admission data, and shift data are never materialized inside the sandbox. The primary final audit covers 60 cases, six reported methods per case, 125 agent rounds, 65 active rounds, and 1,560 selected active rows. A unit test verifies that changing hidden pool outcomes cannot change the selected query indices.

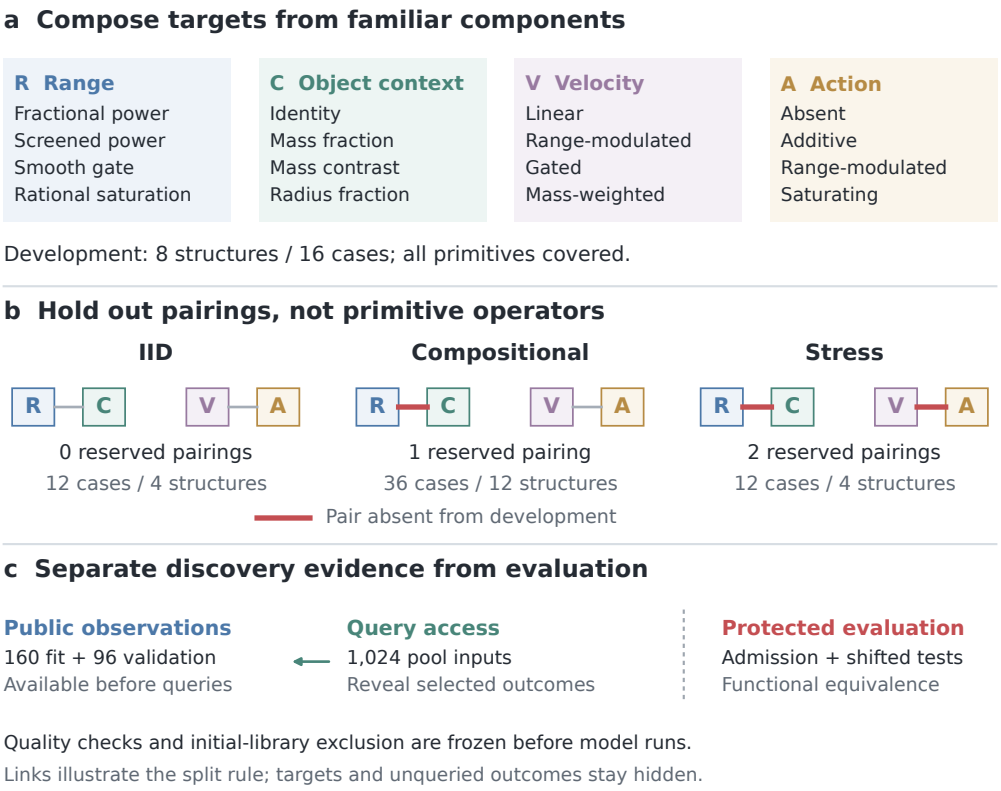


Figure 5: **NovelLaw tests unseen pairings of familiar equation components.** (a) Four component families define the target structures. (b) Red links schematically mark pairings absent from benchmark development; they are not equation-tree edges. (c) Public observations and queried outcomes support discovery, while protected sets evaluate admission, shift, and functional equivalence. Per-case public validation is distinct from the 16 benchmark development cases.

A.2 ALDER Constructs and Revises Equations Beyond Its Initial Set

A.2.1 NovelLaw Protocol

NovelLaw creates structural novelty by recombining four interpretable equation components. Targets are composed from range, object-context, velocity, and action response components. Range is fractional-power, screened-power, smooth-gated, or rational-saturating; context is identity, mass fraction, mass contrast, or radius fraction; velocity response is linear, range-modulated, gated, or mass-weighted; and action is absent, additive, range-modulated, or saturating. Every primitive component occurs in the 16 benchmark development cases. A frozen combinatorial split then reserves selected range–context and velocity–action pairings. The IID split reuses four development motifs with new coefficients and observations; the compositional split contains 12 motifs with one held-out pairing; and the stress split contains four motifs with two held-out pairings. Three independently sampled cases per test motif produce 60 final cases. Full compositional and stress motif signatures are disjoint from development. Figure 5 summarizes the construction and the separation of discovery evidence from evaluation.

The frozen split reserves specific cross-component pairings rather than unseen primitive operators. The reserved range–context pairs are power–mass-contrast, screened–radius-fraction, gated–mass-fraction, and rational–identity. The reserved velocity–action pairs are linear–range-modulated, range-modulated–saturating, gated–absent, and mass-weighted–additive.

NovelLaw separates narrow public evidence from wider protected and functional evaluation domains. Public fit data cover a narrow domain, with independent public validation data for each

Table 5: **Closed-loop revision improves functional recovery and shifted prediction over passive baselines.** The full controlled comparison supplements Table 1A. NovelLaw reports median NMSE over 60 cases; robot errors are mean ± SD over five episode-disjoint partitions. Rejected runs are included. Bold marks the best result in each column, including ties.

	NovelLaw (60 cases)		PullCubeTool (5 runs)		PushT (5 runs)	
Method	Functional recovery (%, ↑)	Shift NMSE (↓)	Admitted (↑)	Shift NMSE (↓)	Admitted (↑)	Shift NMSE (↓)
Baselines						
Fixed/simple model	0.0	5.94×10^{-1}	0/5	5.773 ± 0.287	0/5	2.762 ± 1.098
Polynomial SINDy	28.3	1.16×10^{-1}	0/5	16.573 ± 10.539	0/5	18.281 ± 18.708
Matched MLP	0.0	4.20×10^{-1}	0/5	5.570 ± 0.407	1/5	1.815 ± 1.609
Variants						
One-shot agent	43.3	6.54×10^{-2}	0/5	4.816 ± 1.973	3/5	0.842 ± 0.644
Revision-only agent	75.0	1.50×10^{-2}	0/5	1.296 ± 0.024	3/5	1.292 ± 1.638
Random-query agent	**85.0**	9.15×10^{-3}	**5/5**	0.194 ± 0.138	**4/5**	0.673 ± 0.473
ALDER	**85.0**	$\mathbf{5.58\times10^{-3}}$	**5/5**	$\mathbf{0.161 \pm 0.052}$	**4/5**	$\mathbf{0.420 \pm 0.185}$

Admitted counts use shared predictive checks, not control successes; formula proposals also require structural validity. The first three baselines use initial public evidence. "Fixed" denotes the initial NovelLaw library and a passive linear robot model.

Table 6: **NovelLaw tests new compositions of familiar primitives.** Each test structure has three independently sampled parameter/data instances.

Split	Structures	Cases	Structural role
Development	8	16	Covers every primitive component; used only for protocol and prompt development.
IID	4	12	Reuses a development motif with unseen coefficients and observations.
Compositional	12	36	Contains exactly one reserved component pairing.
Stress	4	12	Contains two simultaneous reserved component pairings.

case. Admission widens interpolation; shift changes distance, mass, radius, velocity, and action ranges. Gaussian noise has scale 0.5% of the fit target standard deviation. A noiseless 4,096-point test set evaluates functional equivalence. Generator draws are retained only after frozen finiteness, variance, component-identifiability, complexity, oracle-refit, and initial-library-exclusion checks. Method outputs are never used for selection.

All NovelLaw baselines share public evidence but differ in how they obtain equation structure. The fixed library has six structures. Polynomial SINDy uses a degree-three library with its threshold selected on public validation data. The MLP has two hidden layers and uses public validation data for early stopping. One-shot and closed-loop methods share the same first proposal; only the latter receives protected rejection feedback and target-blind selected observations. The oracle receives the target structure but must refit its coefficients. A public release contains fit/validation evidence and commitments but excludes targets, protected outcomes, functional-test rows, and unqueried interactive outcomes.

The benchmark checks confirm data separation and method-independent case selection. All 76 generated cases are rechecked against the predefined quality criteria, with no overlapping inputs across data splits. Seventy-one cases pass on their first generator draw; the remaining five require at most three retries. The median oracle functional-test NMSE is below 6×10^{-7}, whereas the median best initial-library NMSE is 0.274. Case selection never reads ALDER outputs. The public release includes public data for all 76 cases but excludes harness-held and evaluator-only data.

A.2.2 AGENT VARIANTS AND FAILURE ANALYSIS

Auxiliary agent variants measure discovery-loop sensitivity but do not isolate the query rule. Revision-only, random-query, and ALDER use separate sandboxes and independent model calls with the same proposal and interaction budgets. One-shot is instead the stored first proposal from the

paired ALDER run. Table 7 therefore measures end-to-end sensitivity to the discovery loop, while RQ2 supplies identical candidate sets to isolate acquisition strategy.

Table 7: **Revised proposals recover more equations than one-shot proposals.** RQ1 values are functional recovery (%); queries are medians over all 60 cases. Independent model calls make the last three rows an auxiliary sensitivity analysis, not a controlled selector ablation.

Method	IID	Comp.	Stress	All	Shift NMSE	Queries
One-shot	41.7	50.0	25.0	43.3	0.0654	0
Revision-only	58.3	86.1	58.3	75.0	0.0150	0
Random-query	91.7	91.7	58.3	85.0	0.00915	24
ALDER	91.7	91.7	58.3	85.0	0.00558	24

Table 8: **Paired tests support gains over one-shot and revision-only discovery.** RQ1 shift-error ratios below one favor closed-loop ALDER.

Comparator	Median shift ratio	Wilcoxon p	Recovery wins/losses
One-shot agent	0.0782	5.18×10^{-9}	25/0
Revision-only	0.146	2.66×10^{-7}	7/1
Random-query	0.901	0.203	3/3
Fixed library	0.00782	1.63×10^{-11}	51/0
Polynomial SINDy	0.0249	2.32×10^{-10}	34/0
MLP	0.0130	1.80×10^{-11}	51/0

Post-hoc failures concentrate in deeper compositions and one repeated gated-range recipe. Recovery is 15/15 for short targets, 21/24 for medium targets, and 15/21 for deep targets. All three instances of the held-out gated-range plus mass-context recipe fail, and the analogous dual-composition gated stress recipe also fails. This pattern is consistent with the composition-depth boundary in the main result. It is descriptive rather than causal because complexity tier and split are not independently randomized.

A.2.3 ROBOT INTERACTION PROTOCOLS

The hard robot tracks use public environments and official demonstrations, but the intervention protocols and evidence partitions are constructed by us. For PullCubeTool, the experiment harness restores a complete Panda–tool–cube state immediately before cube motion, translates only the cube in the L-shaped tool's frame, and replays the identical recorded action segment. Inputs contain the initial relative pose, local tool and cube velocities, relative axis, and commanded tool displacement; the target is cube displacement along the pull axis. Across the five formal folds, 340 source episodes are assigned without overlap to public, query, admission, and shifted roles. Public examples mostly show successful transfer, whereas harness-held interventions include full contact, partial contact, and misses at unseen offsets. Figure 6 contrasts the two robot intervention protocols.

A PullCubeTool query reveals an incorrect contact gate and drives equation revision. Figure 3 illustrates a PullCubeTool run. The lateral coordinate y is measured from the tool-handle centreline to the cube centre. Transferred fraction divides cube displacement by commanded pull remaining after closing the estimated initial gap; plotted curves include fitted transfer gains. Initial evidence lies at 5.8–7.1 cm. All eight queries are shown, some overlapping, and the gray dashed line is a no-gate reference. Near 10 cm, the initial zero-transfer prediction conflicts with near-unit observed transfer. Revision lowers protected-admission NMSE from 2.26 to 0.103; shifted-test NMSE is 0.138. The initial 8.08 cm zero-response endpoint and revised 11.76 cm transition midpoint have different definitions and are not directly comparable thresholds. The lower panels are separate top-view schematics, not trajectory replays; the displayed formulas show only the lateral gates, not the complete prediction equations.

PushT interventions vary contact geometry while replaying the same action to reveal rotational response. For PushT, the experiment harness restores an official ManiSkill Panda state near contact, translates or rotates only the non-convex T, and replays the same action. A paired no-object replay

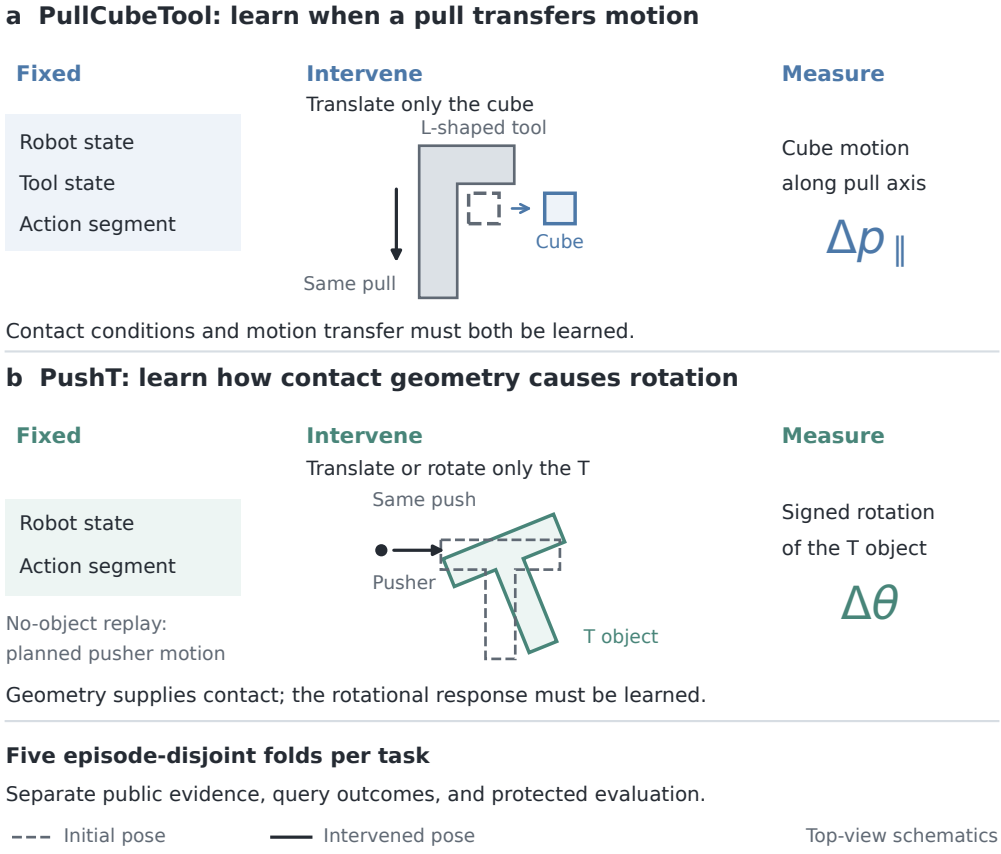


Figure 6: **Matched-action interventions probe contact transfer and rotational response.** (a) PullCubeTool changes only the cube position and measures motion along the pull axis. (b) PushT changes only the T pose and measures signed rotation; geometry supplies the contact input, while a no-object replay supplies planned pusher motion. Both retain the initial robot state and recorded action segment. Outlines show alternative initial poses, not measured trajectories.

supplies intended TCP displacement without revealing the contact response. Inputs contain pusher pose in the initial T frame, the paired TCP displacement, and a swept-contact predicate computed from the known T geometry; the target is signed T rotation. The five folds use 362 disjoint source episodes. Public evidence emphasizes weak contacts, while queries and protected splits vary approach direction, T orientation, and off-center lever arm.

The passive difficulty checks verify both ambiguity and representability before any proposal-agent result is considered. Table 9 shows that a linear model, polynomial SINDy, and matched MLP extrapolate poorly from public evidence. A harness-supplied equation structure also remains ambiguous when its coefficients are fitted only on public rows, but becomes accurate after receiving the full query pool. These supplied structures are benchmark diagnostics and are never presented to the proposal agent.

Each formal agent grid contains the same four conditions and five folds. One-shot receives only the initial public evidence; revision-only receives the same rejection ledger but no new outcome; random-query and ALDER may reveal at most the same number of intervention outcomes. All use the same typed grammar, numerical fitter, proposal budget, admission thresholds, and protected splits. All 40 runs request `codex-default` with high reasoning effort, and each manifest records its CLI version and requested model setting. Table 10 retains rejected runs in every mean.

Interactive revision improves model admission and shifted prediction relative to non-interactive baselines. Across these two tracks, ALDER passes 9/10 paired folds, compared with 3/10 for one-shot

Table 9: **Query evidence improves prediction even when structure is supplied.** Hard-track entries are mean shifted NMSE over five folds; lower is better. "All queries" uses every initially unrevealed query outcome and is not an ALDER discovery result.

Model and evidence	PullCubeTool	PushT
Linear, public	5.773	2.762
Polynomial SINDy, public	16.573	18.281
MLP, public	5.570	1.815
Supplied structure, public	4.579	0.919
Supplied structure, all queries	**0.212**	**0.588**

Table 10: **Interactive revision improves model admission on both robot tracks.** Values are mean $\pm$ standard deviation over five folds. MAE is in centimeters for PullCubeTool and radians for PushT.

	PullCubeTool			PushT		
Method	Admitted	Shift NMSE	Shift MAE	Admitted	Shift NMSE	Shift MAE
One-shot	0/5	4.816 ± 1.973	1.081 ± 0.368	3/5	0.842 ± 0.644	0.086 ± 0.032
Revision-only	0/5	1.296 ± 0.024	0.377 ± 0.032	3/5	1.292 ± 1.638	0.106 ± 0.072
Random-query	**5/5**	0.194 ± 0.138	0.144 ± 0.064	**4/5**	0.673 ± 0.473	0.078 ± 0.021
ALDER	**5/5**	$\mathbf{0.161 \pm 0.052}$	$\mathbf{0.136 \pm 0.029}$	**4/5**	$\mathbf{0.420 \pm 0.185}$	$\mathbf{0.071 \pm 0.024}$

and revision-only. Against either non-interactive condition it has six admission wins, four ties, and no losses (two-sided exact $p = 0.0312$), while its shifted NMSE is lower on all ten folds ($p = 0.0020$). Random-query also passes 9/10; ALDER wins six and loses four shifted-NMSE comparisons against it ($p = 0.7539$). Since random-query and ALDER use independent proposal calls, this grid tests the complete discovery loop rather than the isolated selector. RQ2 supplies that controlled comparison. An independent audit checks all 40 method–fold runs and 76 proposal rounds: every agent executes inside the declared Bubblewrap namespace, no protected or harness-held file appears in its repository, and no failed proposal is removed.

A.3 ARS REDUCES MEAN IDENTIFICATION COST

A.3.1 CONTROLLED MODEL IDENTIFICATION

The intervention protocol isolates information acquisition with matched candidate models, query pools, and budgets. The candidate models vary contact gating, mass–action scaling, action response, and coupled contact response. The safe query pool spans distance, radial speed, action, and mass ratio. It contains geometrically novel points that are irrelevant to the candidate disagreement so that geometric coverage is nontrivial. Candidate coefficients are refitted after each observation. Passive sampling stays near the initial observation regime. Correct identification must be sustained at high confidence. The identification-cost score is the first query count satisfying that criterion for a successful case; an unresolved case receives nine, one above the eight-query budget. All means, medians, and paired reductions use this same score, including failures. They measure penalized identification cost rather than the total number of queries executed. The oracle uses the target model index to choose a query that separates it from alternatives; this greedy reference is not a theoretical lower bound on identification cost.

Identification curves measure retrospectively sustained confidence, not online stopping times. Figure 4 plots the empirical cumulative distribution of sustained identification times. Every curve uses the same 182 initially ambiguous family–seed cases; unresolved trials remain in the denominator at every query count. A time is the first correct, high-confidence round after which every remaining round is also correct at high confidence. All trials execute the full eight-query budget, so these retrospective times do not describe an online early-stopping policy. The curves are reconstructed from the frozen round-by-round confidence records without interpolation. Table 11 retains the complete aggregate comparison, and Table 12 reports its paired uncertainty estimates.

Table 11: **ARS identifies all targets with the lowest mean penalized cost.** RQ2 uses 182 initially ambiguous cases, the same candidate models, and an eight-query budget. Bold marks the best non-oracle results, including ties.

Acquisition	Identified (↑)	Mean cost† (↓)	Median cost† (↓)
Passive sampling	4.4%	8.85	9
Random selection	94.0%	3.53	3
Geometric coverage	**100.0%**	1.91	**1**
Parameter uncertainty	99.5%	1.35	**1**
ALDER (ARS)	**100.0%**	**1.16**	**1**
Oracle reference	100.0%	1.18	1

† **Identification cost includes a penalty for unresolved cases.** Cost measures retrospective sustained identification, not total executed queries; unresolved cases receive nine (budget plus one). The oracle is a reference, not an optimal-cost bound. Paired results appear in Table 12.

Table 12: **ARS reduces mean identification cost relative to each listed comparator.** Paired RQ2 reductions are positive when they favor ARS.

Comparator	Mean cost reduction [95% CI]	Win/tie/loss
Passive	7.69 [7.55, 7.80]	182/0/0
Random	2.36 [2.05, 2.68]	151/29/2
Coverage	0.74 [0.56, 0.94]	63/118/1
Parameter uncertainty	0.18 [0.08, 0.32]	18/161/3

A.3.2 RoboCasa Selector Comparison

The RoboCasa selector comparison varies acquisition while holding candidate equation structures fixed. It uses the worlds specified in Appendix A.4. Table 13 shows that querying is essential for Coulomb and mixed worlds, but does not establish universal superiority of prediction disagreement over coverage or uncertainty. All active selectors eventually identify every Door world, while random selection uses a median of three queries compared with 1.5 for the structured selectors.

Table 13: **Structured queries identify Door models sooner than random queries.** RoboCasa candidates are fixed. Queries are censored at seven after exhausting the six-query budget. "Support" reports exact recovery of the optional equation terms after coefficient thresholding.

Task	Selector	Identified	Support	Median queries	Safety
Drawer	Passive	50.0%	50.0%	3.5	0
	Random	100.0%	97.5%	0.5	0
	Coverage	100.0%	70.0%	0.5	0
	Uncertainty	100.0%	**100.0%**	0.5	0
	ALDER	100.0%	97.5%	0.5	0
Door	Passive	25.0%	25.0%	7.0	0
	Random	100.0%	100.0%	3.0	0
	Coverage	100.0%	100.0%	**1.5**	0
	Uncertainty	100.0%	100.0%	**1.5**	0
	ALDER	100.0%	100.0%	**1.5**	0

A.4 Equation-Based Models Improve Shifted Prediction and Support Control

A.4.1 Environments and Evidence Collection

The robot study uses physically reached states and matched worlds within each articulation. The end-to-end robot study uses RoboCasa 1.0.1 at commit `a07e365c` and robosuite 1.5.2 at commit `5ce6643f`. We instantiate the prismatic `OpenDrawer` and revolute `OpenDoor` tasks with the

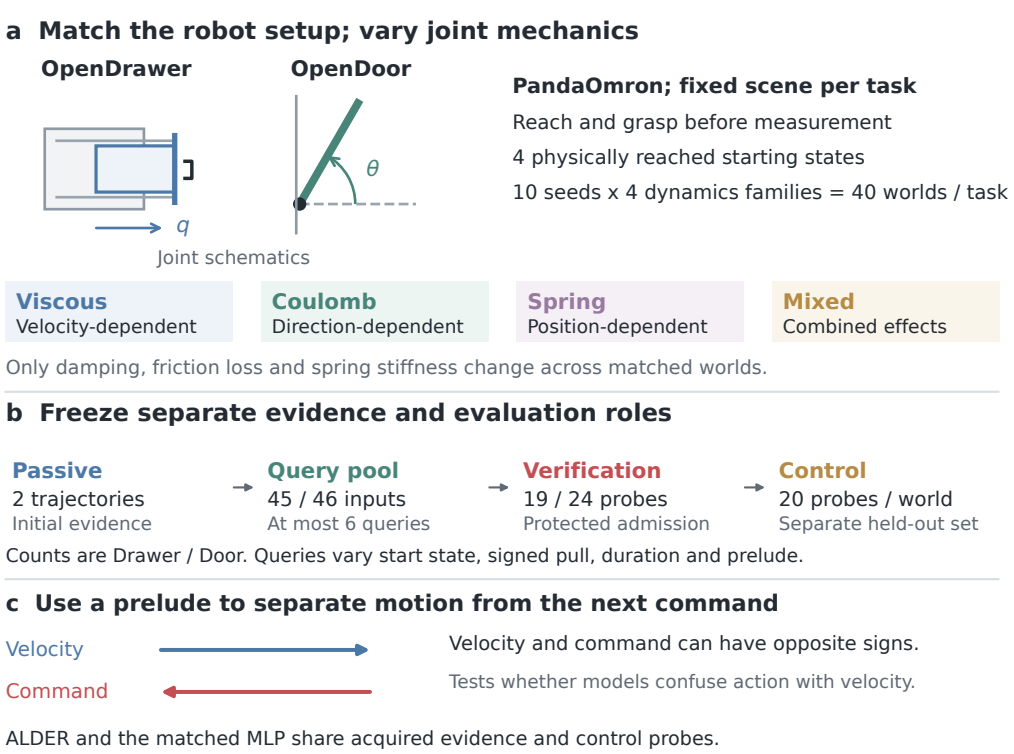


Figure 7: **Matched RoboCasa worlds isolate joint mechanics and separate discovery from control evaluation.** (a) Joint schematics illustrate the two post-grasp tasks and four dynamics families. (b) Passive, query, admission, and control probes have separate roles; paired counts denote Drawer and Door. (c) A prelude can make motion oppose the next command, exposing explanations that substitute action direction for velocity direction. Arrows illustrate this intervention, not a measured rollout.

same PandaOmron robot, a 20 Hz controller, state observations, and no rendering. A deterministic common controller reaches and grasps each handle, physically visits four normalized articulation positions, and saves the resulting MuJoCo states. Every scored rollout begins from a reached state; direct teleportation is not used to construct a grasp. Drawer evaluation seeds are 1701–1710 and Door seeds are 2701–2710. Within an articulation, only the hidden joint dynamics change across matched worlds.

Both articulations use matched dynamics families and a shared evidence protocol. Each articulation contains 40 worlds: ten seeds for viscous damping, Coulomb friction, linear spring stiffness, and their composition. Every world has two narrow passive trajectories, a safe query pool, protected admission probes, and 20 disjoint control probes. Drawer provides 45 query candidates and 19 admission probes; Door provides 46 and 24. A nominal-world screen fixes legal probe IDs before any shifted outcome is evaluated. Queries vary initial position, signed pull, horizon, and an optional prelude that can make velocity direction disagree with action direction, testing whether models confuse these directions when predicting friction. All methods receive the same cached outcomes and may request at most six queries. Figure 7 summarizes the matched worlds and probe roles.

A.4.2 Equation Discovery and Control Evaluation

Joint-force equation discovery expands a simple initial program through bounded structural proposals and separate coefficient fitting. The initial program has bias and velocity terms. In each round, the proposal agent may add at most two terms from a declared typed grammar, numerically compare at most four self-proposed structures, and submit one structure fitted on public evidence.

Table 14: **Equation-based models improve prediction under the original finite-action evaluation.** Values are mean ± SD over ten seeds, each averaging four dynamics families. Relative success means executed target error within 0.01 of the best finite candidate's error; it is not the absolute target-reaching success in Table 2. Each task has 640 targets per model. Bold marks the best non-oracle results, including ties.

	OpenDrawer		OpenDoor	
Model	OOD prediction RMSE (↓)	Relative success (%, ↑)	OOD prediction RMSE (↓)	Relative success (%, ↑)
Nominal	0.0068 ± 0.0007	96.7 ± 1.4	0.0072 ± 0.0008	95.9 ± 2.0
Passive law	0.0075 ± 0.0010	94.2 ± 2.1	0.0074 ± 0.0015	96.4 ± 2.4
Matched MLP	0.0037 ± 0.0008	99.5 ± 1.1	0.0031 ± 0.0007	$\mathbf{100.0 \pm 0.0}$
ALDER law	$\mathbf{0.0009 \pm 0.0001}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{0.0009 \pm 0.0001}$	$\mathbf{100.0 \pm 0.0}$
Parameter oracle	0.0009 ± 0.0001	100.0 ± 0.0	0.0009 ± 0.0001	100.0 ± 0.0
Outcome oracle	0.0000 ± 0.0000	100.0 ± 0.0	0.0000 ± 0.0000	100.0 ± 0.0

Candidate primitives include position, friction sign, smooth sign, speed magnitude, quadratic velocity or position, action interactions, and an exponential position feature. The complete reference equation is absent from the sandbox. Harness-owned ridge fitting estimates coefficients, and held-out normalized error controls admission. The primary proposal backend is `gpt-5.6-sol` with frozen reasoning settings. The same generic joint interface and proposal schema are used for Drawer and Door.

The original finite-action comparison shares acquired evidence between the MLP and ALDER. For control, the passive law is fitted only to passive evidence, whereas the matched MLP and ALDER law use exactly the passive plus ALDER-selected evidence. The MLP has two 32-unit tanh layers and fixed regularization. Every predictor forecasts the same frozen control probes. For each staged state, the planner forms targets at 20, 40, 60, and 80 percent of the reachable articulation range and selects the action whose predicted final position is closest. Relative success means that executed error is within 0.01 normalized position of the best available candidate. A rejected model refuses all targets in that world and receives zero relative control success. The parameter oracle knows hidden joint coefficients but not cached outcomes; the outcome oracle directly ranks those outcomes. Each task contains 3,840 matched model–target rows across the six control models.

The original RoboCasa uncertainty estimates use paired seeds, with failed proposals and prediction coverage explicitly accounted for. All confirmatory summaries use the ten seeds within each articulation as the paired bootstrap unit with 10,000 resamples. Admission and term-support recovery retain every failed proposal in the denominator. Continuous prediction metrics are conditional on both compared methods producing finite predictions and are reported with coverage.

Numerical inverse control tests the learned models against absolute target-reaching criteria. The extension retains all 80 worlds and 1,280 targets, with discovered equations, admission decisions, and training evidence unchanged. All model-based methods use the same bounded Brent solver, action ranges, and horizon order (11, 17, then 7 steps), with at most 48 forward-model evaluations per target. A root must have predicted target error at most 10^{-4} and satisfy predicted safety constraints; otherwise, the solver returns the closest safe evaluated action or refuses execution. Actual outcomes are evaluated only after action selection. Table 2 retains prediction RMSE on the original fixed OOD probes and adds mean absolute executed endpoint-to-target error and success under numerical inversion. Success requires actual target error at most 0.01 and safe execution; refusals count as failures. Means and sample SDs use ten seeds per task, each averaging four families and 16 targets per family. These success rates are not directly comparable to the original relative-success metric.

The known-parameter model remains a reference under the same inverse solver. It achieves $100.0 \pm 0.0\%$ success on both tasks, with control MAE $(0.74 \pm 0.10) \times 10^{-3}$ on Drawer and $(0.50 \pm 0.06) \times 10^{-3}$ on Door. The outcome oracle remains confined to the finite-action comparison in Table 14; it is not an optimum over the continuous action range.

The complete agent loop transfers without changing its interface between the two articulations. Across the 40 worlds in each task, every submitted ALDER model passes protected admission. Strict

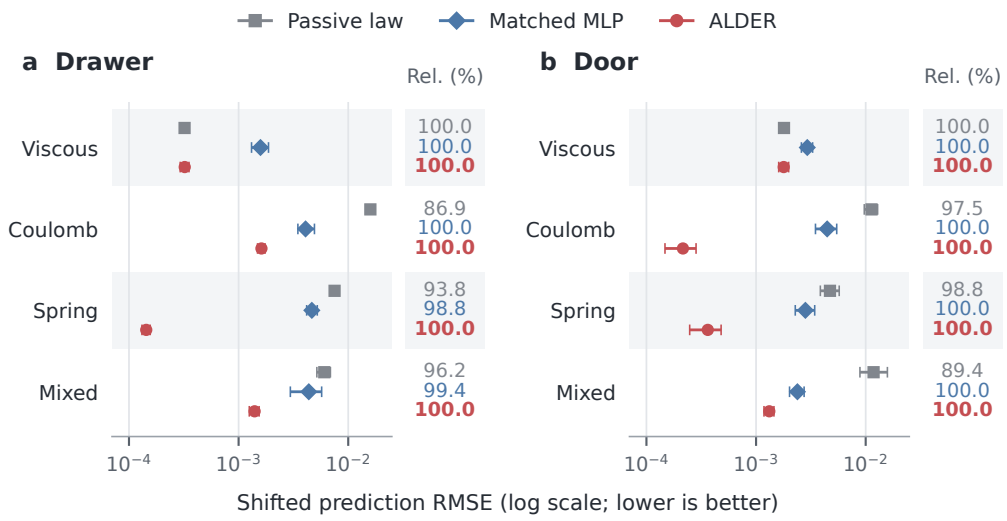


Figure 8: **ALDER predicts more accurately than the MLP in every tested robot family.** Points show mean shifted prediction RMSE on a shared logarithmic scale; whiskers are 95% percentile-bootstrap intervals over ten matched seeds (10,000 resamples). The adjacent columns retain relative finite-action success for each method in the same row order.

term-support recovery is 85.0% on Drawer and 95.0% on Door, and both tasks use a median of 0.5 new queries. The geometric mean protected NMSE is 5.65×10^{-8} and 1.92×10^{-7}, respectively. These rates evaluate open structural proposal; Appendix A.3.2 separately isolates acquisition using fixed candidate equation structures.

A.4.3 Dynamics-Family Results

ALDER reduces prediction error in every tested task–family cell, while relative finite-action success is largely saturated. Figure 8 reports every registered family for both articulations under the original finite-action evaluation. The updated frozen reproduction gives ALDER lower prediction RMSE than the matched MLP in all eight task–family cells, while relative control success is already saturated in most cells. We therefore emphasize prediction under dynamics shifts rather than a large control-success advantage.

A.5 Verification Rejects Models Unsupported by Available Evidence

A.5.1 Missing Structure and Hidden State

The equation-and-residual benchmark varies model completeness while matching observed inputs, noise, sample counts, and domains. Every regime shares observed distance, speed, action, mass, noise standard deviation 0.03, sample counts, and ID/shift domains. The known six-term library contains a constant, distance offset, speed, action divided by mass, a smooth contact-gated action-over-mass term, and inverse-square distance. Missing terms are never given to the learner. For incomplete-grammar tests, the residual and gate are trained only after equation fitting, and boundary labels are evaluated only by the evaluator.

Model admission and residual localization use separate frozen criteria. The equation coefficients are fitted by ridge regression. A two-layer SiLU MLP predicts either the full target or the residual. A second MLP classifies pseudo-boundary labels formed when the absolute training residual exceeds four noise standard deviations. Admission uses held-out ID validation data, requiring equation-only NMSE at most 0.02 and mean gate probability below 0.10; shifted data serve only for evaluation.

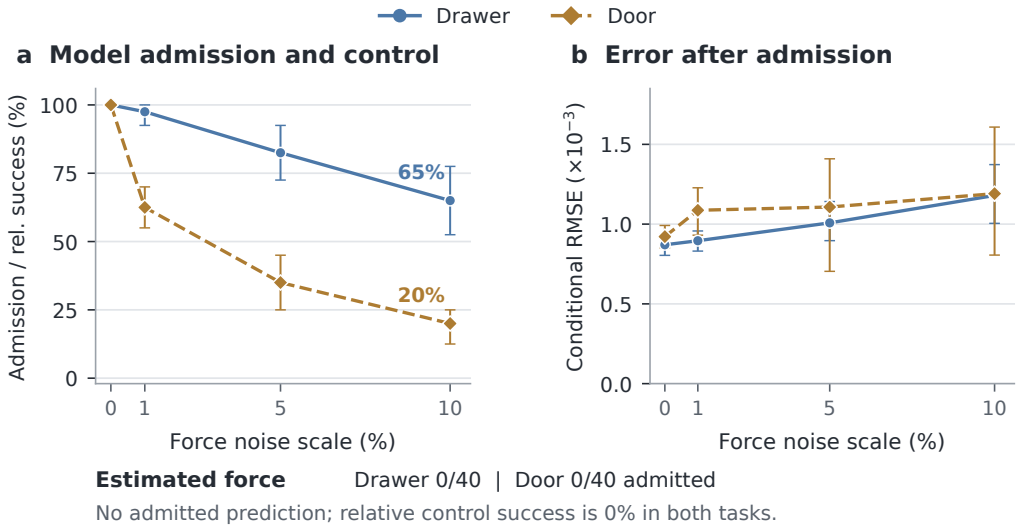


Figure 9: **Observation noise reduces admission while retained predictions remain accurate.** Each task contains 40 worlds across ten seeds. Points show observed means; whiskers are 95% seed-bootstrap intervals (10,000 resamples). Zero noise denotes the force reference. Control uses the original finite-action relative-success criterion (Table 14), not the inverse-control criterion in Table 3B. Relative success equals admission in these runs; rejected worlds have no prediction and receive zero control success. RMSE is conditional on admission. Force estimation from motion and known actions (labeled "Estimated force") is evaluated separately, not as another noise level.

Table 15: **Residual correction helps missing-term regimes but not hidden-state failures.** RQ4 reports shifted prediction NMSE; lowest error is bold.

Regime	Equation only	MLP	Always residual	Gated residual
Complete	4.20×10^{-5}	0.0109	0.000345	4.20×10^{-5}
Missing smooth	0.217	0.0133	**0.00380**	0.0145
Missing local	0.0485	0.0308	0.0191	**0.0185**
Hidden local	**0.108**	0.134	0.183	0.168

A.5.2 Observation-Channel Stress Tests

The observation-channel stress test refits the validated equation structures after degrading their target signal. This stress test uses the RoboCasa worlds and control protocol specified in Appendix A.4. Noise conditions add 1%, 5%, or 10% scale noise plus a fixed per-rollout bias. The estimated-force condition replaces direct force observations with estimates from motion and known control actions. It estimates nominal acceleration with a 300-tree extra-trees regressor over position, velocity, and action, then applies a task-specific known-force calibration. Figure 9 reports every task and channel. The test is intended to expose an observability boundary, not to guarantee successful validation with estimated forces.

The main observation-stress results use the same numerical inverse solver and absolute success criterion as RQ3. Table 3B retains each channel's fitted equations, admission decisions, and fixed-probe prediction RMSE. Control success is evaluated over all targets, counting refusals and unsuccessful executions as failures. It is distinct from the relative finite-action success shown in Figure 9.



Figure 10: **Closed-loop discovery improves recovery, identification, and shifted prediction.** (a) Iterative revision improves NovelLaw recovery over the first proposal. (b) Prediction disagreement reduces retrospective identification cost with fixed candidate models. (c) Admitted models reduce shifted prediction error on two RoboCasa articulations. (d) Verification rejects all tested incomplete models; the auxiliary localization diagnostic remains near chance for the shifted missing-smooth regime. Error bars show 95% bootstrap intervals for functional recovery in (a) and paired identification-cost reductions in (b).

A.6 Equation Discovery Extends Across Systems and Proposal Backbones

A.6.1 External ODE Benchmarks

The external ODE evaluation checks whether equation construction extends beyond the dynamics families designed for our main experiments. We execute ALDER on all 63 ODEBench systems (d'Ascoli et al., 2024) and the 59 valid two- and three-dimensional ODEBase systems (Lüders et al., 2022) used by LLM-ACES (Abhyankar et al., 2026). The proposal agent receives anonymized public state and derivative observations; generalization trajectories and extended-horizon trajectories remain evaluator-held until final evaluation. Tables 16 and 17 report median normalized mean squared error (NMSE), mean expression complexity, and symbolic accuracy.

Quartiles describe variation across ODEBase systems, not repeated runs. For ALDER, the NMSE [Q1, Q3] intervals are [9.49e-19, 6.19e-14] for reconstruction, [7.68e-19, 2.19e-14] for generalization,

and [9.88e-32, 1.20e-23] for future-time OOD. These use linear interpolation across all 59 systems, including the preset 10^{30} penalties for non-finite errors.

ALDER obtains low median error across reconstruction, unseen initial conditions, and extended horizons. A separate, stricter functional test recovers 60/63 ODEBench systems and 49/59 ODEBase systems; this count is not the symbolic-accuracy metric in the tables. It requires finite trajectory NMSE at most 10^{-6} in all three prediction settings. These results show that the harness can execute across broad ODE collections.

Table 16: **ALDER achieves low median error across ODEBench prediction settings.** We report the median NMSE (lower is better) across reconstruction, generalization, and future-time OOD prediction, mean symbolic accuracy (higher is better), and mean expression complexity (ODEBench mean = 19.3). Best NMSE and symbolic-accuracy results are in bold, and second-best results are underlined. The ALDER row is shaded.

Method	Recon. NMSE ↓	Gen. NMSE ↓	OOD NMSE ↓	Complexity	Sym. Acc (%) ↑
Passive Symbolic Discovery					
SINDy	5.07e-04	5.09e-01	1.41e+00	11.7	18.5
Operon	7.95e-05	2.57e-01	1.84e+00	14.0	2.4
PySR	2.84e-03	8.82e-01	1.43e+00	6.6	18.1
E2E	3.69e-01	1.49e+00	2.19e+00	52.6	0.0
ODEFormer	4.61e-03	3.83e-01	2.04e+00	15.9	16.5
LLM-guided Symbolic Discovery					
LLM-only	2.20e-08	3.45e+00	1.79e+00	38.9	0.0
LLM-ODE	4.12e-05	4.72e-03	2.43e-02	25.2	5.9
Active Symbolic Discovery					
Query-by-Committee (QBC)	1.81e-09	5.07e-09	5.69e-08	33.7	15.6
Bayesian Optimization (BO)	1.06e-14	9.35e-13	3.47e-10	22.6	41.2
APPS-ODE	7.52e-01	8.13e-01	1.02e+00	13.4	2.6
LLM-ACES (GPT)	**1.33e-17**	**8.28e-17**	2.46e-16	17.1	46.2
LLM-ACES (Qwen)	6.30e-16	4.18e-15	1.89e-15	18.2	45.6
ALDER	1.98e-15	1.07e-15	**4.01e-29**	17.5	**95.0**

Table 17: **ALDER maintains low median prediction error on ODEBase.** We report the median NMSE (lower is better) across reconstruction, generalization, and future-time OOD prediction, mean symbolic accuracy (higher is better), and mean expression complexity (ODEBase mean = 35.2). Best NMSE and symbolic-accuracy results are in bold, and second-best results are underlined. The ALDER row is shaded.

Method	Recon. NMSE ↓	Gen. NMSE ↓	OOD NMSE ↓	Complexity	Sym. Acc (%) ↑
Passive Symbolic Discovery					
SINDy	1.06e-03	1.12e+00	3.47e-01	12.5	5.9
Operon	1.49e-05	1.46e+00	1.41e+00	19.0	5.4
PySR	1.37e-03	1.08e+00	1.35e+00	8.8	5.9
E2E	1.04e+00	1.19e+01	1.98e+01	83.6	0.0
ODEFormer	1.64e-02	1.15e+01	4.09e+00	18.5	4.8
LLM-guided Symbolic Discovery					
LLM-only	4.52e-08	1.06e+01	1.11e+00	54.9	0.0
LLM-ODE	7.46e-05	8.51e-01	6.59e-04	32.1	0.1
Active Symbolic Discovery					
Query-by-Committee (QBC)	4.97e-09	7.47e-06	6.38e-06	45.2	14.1
Bayesian Optimization (BO)	8.55e-10	4.61e-10	5.43e-08	31.3	49.3
APPS-ODE	5.56e-01	9.12e-01	1.01e+00	10.9	15.2
LLM-ACES (GPT)	2.54e-14	8.50e-10	3.05e-12	33.1	50.0
LLM-ACES (Qwen)	3.70e-15	4.18e-15	3.44e-13	31.2	52.4
ALDER	**2.32e-15**	**2.28e-16**	**3.28e-29**	30.6	**90.7**

Reference results come from LLM-ACES, whose symbolic-accuracy evaluation code is not publicly available. All non-ALDER entries are taken from its Tables 2–3 and were not rerun locally. ALDER's symbolic accuracy uses an independently implemented pipeline following the LLM-ACES evaluation procedure and the same GPT-4o mini judge model.

A.6.2 ARTICULATED GEOMETRY AND CONTACT CONSTRAINTS

Door and Peg test articulated geometry and non-smooth constraints without requiring multi-round revision. Every formal Agent run succeeds on its first proposal, so these tasks serve as breadth checks rather than evidence that revision is needed. Adroit Door uses episode-disjoint angle ranges to test additive approximations fitted at low angles. We use the standard Minari

`D4RL/door/human-v2` conversion (Rajeswaran et al., 2018; Fu et al., 2020), containing 25 human demonstrations and 6,729 transitions. For each of five episode partitions, public fit and development rows contain hinge angles no larger than 0.65 rad. Protected ID holds out episodes in the same range; protected OOD holds out episodes at hinge angles of at least 0.90 rad. Inputs are current and initial hinge/latch angles, and the target is handle x displacement. Admission requires development, ID, and OOD NMSE below 0.04, 0.05, and 0.02, respectively, plus at least 90% improvement over the zero-displacement baseline on both protected splits.

Every admitted Door equation contains the composed geometric dependence needed for high-angle extrapolation. The agent cannot see protected rows or the harness-held composed trigonometric basis. Every first proposal passes admission. The admitted equations use hinge sine/cosine displacement together with a latch term rotated by the hinge, rather than adding the two joints independently. Table 18 reports all registered baselines.

Table 18: **One-shot joint equations extrapolate to unseen Door angles.** Adroit Door uses five episode partitions. Errors are handle-x MAE in cm; lower is better. OOD contains only unseen high-angle states.

Method	Public dev.	Protected ID	High-angle OOD	Admitted
Zero displacement	6.671	6.501	28.866	0%
Linear joints	0.810	0.813	11.660	0%
Polynomial SINDy-3	0.025	0.027	1.455	40%
MLP	0.047	0.051	10.546	0%
Additive trigonometric library	0.068	0.065	2.626	0%
Full composed reference	0.020	0.021	0.568	100%
One-shot agent / ALDER	0.046	0.047	**0.339**	100%

Peg interventions test contact constraints beyond correlations with task progress. The 1,000 official motion-planning demonstrations contain 149,055 labeled states, but follow a narrow approach path. A linear classifier obtains about 0.996 balanced accuracy on those states by relying on task progress. We therefore execute 56 orthogonal state interventions for every official episode seed. Seven balanced probe families isolate success, axial failure, positive/negative y failure, positive/negative z failure, and simultaneous failures; half of the relevant probes lie near a boundary. ManiSkill executes all 56,000 probes and supplies the only public target, a binary success label.

Peg evaluation separates central-radius fitting from protected episode and radius shifts. Public fit uses 40 central-radius episodes (2,240 labels), public development uses 20 (1,120), protected ID uses 40 (2,240), and radius OOD uses 80 (4,480) split equally between disjoint small- and large-hole ranges. Admission requires at least 0.98 balanced accuracy and 0.95 success F1 on development, protected ID, and radius OOD. Across five partitions, every first proposal produces an equivalent three-branch constraint and passes admission (Table 19). The continuous simulator margin is never released to the agent or learned baselines.

Table 19: **One-shot constraints generalize across held-out peg-hole radii.** PegInsertionSide provides binary simulator labels. All metrics except "Admitted" are balanced accuracy or success F1; higher is better.

Method	Public dev.	Protected ID	Radius OOD	OOD success F1	Admitted
Always failure	0.500	0.500	0.500	0.000	0%
Linear	0.557	0.553	0.552	0.266	0%
Polynomial logistic-3	0.769	0.767	0.754	0.450	0%
MLP	0.905	0.901	0.796	0.605	0%
Additive absolute rule	0.649	0.646	0.638	0.369	0%
Full non-smooth reference	1.000	1.000	1.000	1.000	100%
One-shot agent / ALDER	**1.000**	**1.000**	**1.000**	**1.000**	100%

A.6.3 PROPOSAL-BACKBONE ROBUSTNESS

The open-weight ablation changes only the proposal backbone while fixing the experimental framework. All models use the same stratified subset of 12 RoboCasa worlds across three paired seeds, with identical public evidence, selector, coefficient fitter, admission criteria, and query budget. Table 20 compares the primary GPT-5.6 backend with seven valid open-weight backbones on this subset. Failed proposals remain in the denominators of admission and term-support recovery rates.

ALDER discovers validated equations with multiple open-weight proposal models. Across the seven open-weight backbones, protected admission ranges from 58.3% to 100.0% and strict term-support recovery from 33.3% to 75.0%. The primary GPT-5.6 backend obtains 100.0% admission and 83.3% term-support recovery on the identical subset. All backbones record zero safety violations. The pattern is not monotonic in total or active parameter count: the dense 27B model has the highest open-weight term-support recovery, whereas the sparse 80B model has the lowest admission.

Table 20: **ALDER discovers validated equations with multiple open-weight backbones.** On 12 matched RoboCasa worlds, the first row is the primary backend; the remaining seven are open-weight ablations. Queries are medians with non-admitted runs assigned budget plus one; NMSE is the geometric mean over finite predictions. Total/active parameter counts follow public model declarations, while the primary parameter count is not public. Tokens are API-reported mean request totals and are not a quality-normalized cost metric. "Support" uses the same strict optional-term recovery criterion as the primary study.

Backbone	Total/active B	Admitted	Support	Queries	Pred. cover	NMSE	Tokens
GPT-5.6 (primary)	not public	100.0%	**83.3%**	0.5	100.0%	7.85×10^{-8}	345788
Qwen3.5-9B	9/9	100.0%	66.7%	1.0	100.0%	3.34×10^{-7}	35082
GPT-OSS-20B	21/3.6	100.0%	41.7%	1.0	100.0%	3.16×10^{-6}	43037
Qwen3.8-27B	27/27	91.7%	75.0%	0.5	100.0%	8.77×10^{-7}	56292
Qwen3.6-35B-A3B	35/3	91.7%	41.7%	1.0	100.0%	3.93×10^{-6}	54832
Qwen3-Coder-Next	80/3	58.3%	33.3%	1.5	100.0%	5.85×10^{-5}	78094
GPT-OSS-120B	117/5.1	100.0%	41.7%	1.0	100.0%	2.67×10^{-6}	31470
Qwen3-235B-A22B	235/22	100.0%	50.0%	0.5	100.0%	3.04×10^{-6}	78166

A.7 STATISTICAL PROTOCOLS AND MODEL SETTINGS SUPPORT REPRODUCIBILITY

A.7.1 STATISTICAL PROTOCOLS

Statistical comparisons preserve the matched unit appropriate to each experiment. All final comparisons are paired by generated case or robot task. Interaction counts use percentile-bootstrap intervals. RQ1 recovery-rate intervals cluster the bootstrap over 20 structure recipes, retaining all three parameter/data instances of a sampled recipe; its log-NMSE comparisons are paired by case. Wilcoxon signed-rank and exact McNemar tests are reported as secondary paired diagnostics.

A.7.2 PROPOSAL MODELS AND RESOURCE USE

NovelLaw's primary proposal backend was identified as GPT-5.6 at evaluation time. The primary runs used the default backend, which resolved to `gpt-5.6-sol` with low reasoning effort; auxiliary revision and random-query runs explicitly specified `gpt-5.6-sol`.

We summarize resource use through token counts and cumulative agent time. Table 21 reports usage over 60 NovelLaw cases per variant; on RoboCasa, mean API-reported usage is 413,528 tokens per Drawer world and 323,520 per Door world.

Table 21: **Each full agent variant uses one invocation per case.** RQ1 usage is summed over 60 cases. One-shot is derived from ALDER's stored first round and incurs no additional call.

Method	Calls	Proposals	Queries	Input tokens	Agent h
Revision-only	60	138	0	31.8M	1.97
Random-query	60	117	1,368	30.9M	2.15
ALDER	60	125	1,560	34.1M	2.22