# PROBE–DIAGNOSE–ADAPT: AN EVIDENCE-DRIVEN EMBODIED RESEARCH LOOP FOR ROBOT CO-DESIGN

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Robot co-design is costly because evaluating a morphology typically requires training a controller for it. Each trial, however, reveals more than a task return: it shows how the body moves and interacts with its surroundings. How can these observations guide the next design change? We introduce *Probe–Diagnose–Adapt* (PDA), an LLM-guided framework that organizes robot co-design as an embodied research loop. PDA trains a fresh controller for each candidate under the same training budget and summarizes its behavior through module-level probes fixed after calibration. Drawing on this evidence and past trials, the LLM identifies a capability to investigate, proposes a feasible body edit, and predicts a measurable change before testing. Each experiment assesses task performance and the predicted change separately: performance guides morphology selection, while the prediction check informs subsequent reasoning. The resulting experimental memory preserves exact edits and both outcomes, allowing successful, rejected, and inconclusive trials to inform later proposals. Under matched per-body training budgets, PDA achieves the highest mean search performance at 53 morphologies on all six *Evolution Gym* tasks. On five tasks, these returns are competitive with the endpoints of longer baseline searches. Controlled ablations show complementary gains from behavioral evidence, module-level localization, and experimental memory. On two articulated-robot tasks in *UNIMAL*, the same loop improves on initial designs, supporting applicability beyond voxel-based morphologies. By linking edits to prospective predictions and measured outcomes, PDA takes a step toward embodied auto research through budgeted experimentation.

## 1 INTRODUCTION

Learning a capable controller is only part of building a robot. A body's geometry, compliance, and actuator arrangement shape its interaction with the environment and can make useful behavior easier or harder to learn. *Robot co-design* therefore optimizes morphology alongside control, rather than treating the body as a fixed platform for policy learning (Gupta et al., 2021; Bhatia et al., 2021; Wang et al., 2025). This coupling also makes co-design expensive: assessing a new body generally requires learning how to control it. Existing approaches tackle this challenge through population-based search, joint learning of design and control, and the reuse of policies and representations across bodies (Sims, 1994; Yuan et al., 2022; Gupta et al., 2022; Wang et al., 2023b). A complementary opportunity is to make better use of each experiment. Beyond the return used to rank designs, rollouts reveal where a body moves, stalls, or interacts with its surroundings. These observations can inform not only which design to retain, but also which intervention to test next.

Consider a robot that repeatedly stalls at a terrain transition. Its task return alone leaves two possibilities unresolved: the body may restrict upward motion, or the controller may not have learned to produce it. To investigate these possibilities, we can use local motion and body–terrain geometry to identify a structural intervention. By stating the expected behavioral effect before making that intervention, the next experiment can ask two distinct questions: under the same learning budget, does the edit improve task progress, and does the intended increase in upward motion occur?

Large language models (LLMs) offer a way to reason about such interventions by combining task knowledge, body layouts, and experimental records. Existing systems use these capabilities for candidate generation, reflection, and experience reuse (Qiu et al., 2024; Qiu & Cygan, 2025; Song et al.,



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

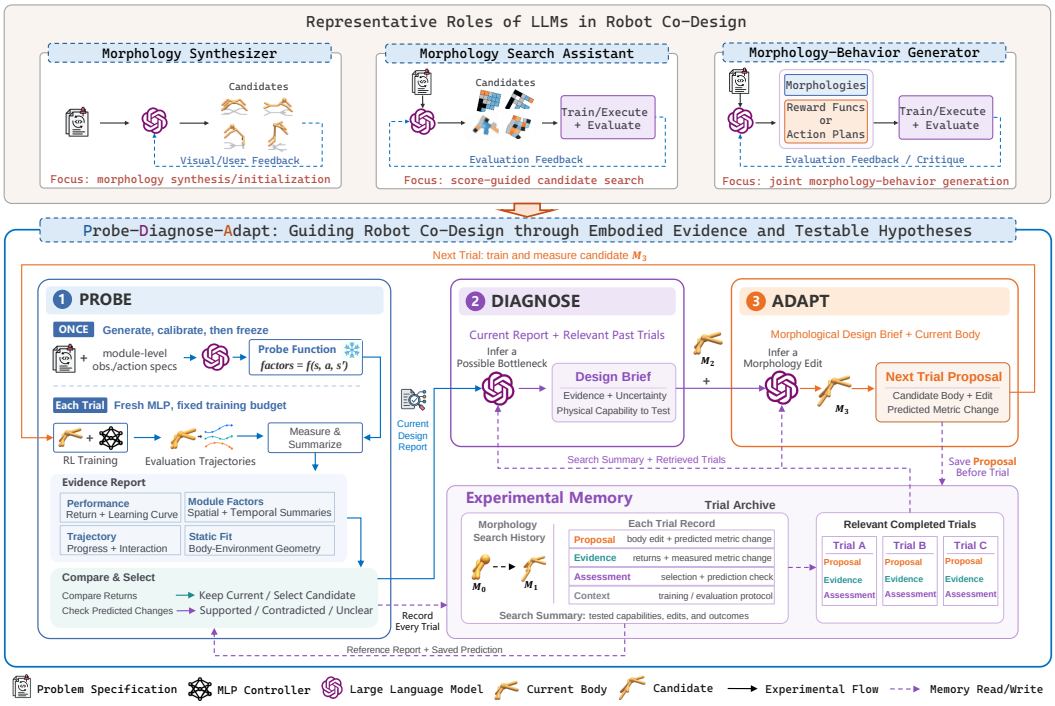


Figure 1: **Overview of Probe–Diagnose–Adapt (PDA). Top:** Three roles of LLMs in robot co-design. **Bottom:** PDA combines behavioral evidence with experimental memory to propose testable body edits, then assesses performance and predicted changes separately to guide subsequent trials.

2025; Wang et al., 2026). Our focus is on connecting this reasoning to a specific experimental test. A behavioral prediction stated in advance helps distinguish an intended change that was not realized from one that occurred without improving task performance. These outcomes suggest different directions for subsequent design, making a trial informative beyond its return. This leads to our research question: *how can an LLM use behavioral evidence from successive trials to make better morphology interventions when only a limited number of bodies can be trained and evaluated?*

To address this question, we introduce *Probe–Diagnose–Adapt* (PDA), an embodied research loop that pairs each proposed body edit with a measurable expectation for the next experiment (Figure 1). At initialization, an LLM generates a shared module-level probe from the task and observation/action specifications; the probe is calibrated and then held fixed throughout the search. For each candidate, *Probe* trains a fresh controller under a fixed budget and combines probe-derived factors with system measurements of performance, trajectory outcomes, and body–environment geometry into a structured evidence report. Module-level measurements connect observed behavior to editable body components, while fixed measurement definitions provide a common basis for comparing trials. Drawing on the evidence and relevant past experiments, *Diagnose* identifies a capability worth investigating and produces a design brief. *Adapt* translates this brief into a feasible candidate body and a predicted observation, then returns the candidate to controller learning and evaluation.

Each trial yields two distinct outcomes: whether the candidate improves its evaluation score and whether it produces the predicted behavioral change. The score guides morphology selection, while the prediction assessment informs how the intervention is interpreted. In the terrain example, failure to increase upward motion suggests revisiting whether the edit and learned controller realize the proposed capability. Increased upward motion without better task progress instead suggests reconsidering whether that capability addresses the task difficulty. Experimental memory binds the exact edit, prior prediction, measured changes, and both outcomes to their experimental context. All completed trials are retained, including rejected candidates and unclear prediction checks. Thus, PDA can revise an explanation while retaining a useful body, or learn from a rejected edit.

PDA seeks high-performing body–controller pairs within fixed training and search budgets. Each candidate receives one controller-training run, followed by evaluation of the selected checkpoint. The score guides selection, while the evaluation trajectories inform the next intervention. With the learner and evaluation protocol held fixed, the challenge is to make each experiment more useful within the remaining budget. As earlier outcomes shape what is tested next, morphology search becomes a cumulative process of embodied experimentation—a step toward *Embodied Auto Research*.

Our contributions are threefold: **(1)** We introduce PDA, an embodied experimentation loop that turns module-level behavioral evidence into testable morphology edits. Calibrated probes connect observations to editable components, allowing the LLM to state a measurable expectation before committing the next controller-training budget. **(2)** We develop experimental memory that links exact edits, prospective predictions, and measured outcomes while distinguishing performance improvement from prediction support. This allows subsequent design to build on successful interventions, revisit unsupported explanations, and reuse evidence from rejected or inconclusive trials. **(3)** With shared controller-training and evaluation protocols, PDA achieves the highest mean search performance at 53 trained morphologies, including three initial bodies, on all six *Evolution Gym* tasks. On five tasks, its returns are also competitive with much longer baseline searches. Information ablations examine the roles of behavioral evidence, localization, and experimental memory, while two *UNIMAL* tasks demonstrate applicability to articulated bodies.

## 2 Related Work

**Robot co-design.** A change in morphology also changes the control problem, so robot co-design methods differ in how they organize structural exploration and policy learning (Liu et al., 2025; Dong et al., 2025; Ikemura et al., 2026; Wang et al., 2025). Population-based approaches evaluate successive body–controller pairs (Sims, 1994; Gupta et al., 2021), including the evolutionary baselines studied in Evolution Gym (Bhatia et al., 2021). Joint RL methods learn design decisions together with control policies (Yuan et al., 2022; Wang et al., 2023a), while policy sharing and pretraining reuse control experience across morphologies (Gupta et al., 2022; Wang et al., 2023b; Strgar & Kriegman, 2026). Complementing policy sharing, PDA guides morphology search by reusing behavioral evidence rather than controller parameters.

**Language-guided morphology design and experience reuse.** Language-guided methods bring task specifications and experimental feedback into robot design. Figure 1 summarizes three representative, overlapping roles. A *morphology synthesizer* generates bodies from specifications, as in RobotDesignGPT's visual and user-guided refinement (Sontakke et al., 2026). A *morphology search assistant* guides candidate exploration, exemplified by RoboMorph's grammar-based evolution (Qiu et al., 2024) and LASeR's diversity-oriented reflection (Song et al., 2025). A *morphology–behavior generator* jointly proposes physical form and behavioral specifications: RoboMoRe optimizes morphology and training rewards (Fang et al., 2025), VLMgineer generates tool geometries and action plans (Gao et al., 2026), and Debate2Create uses simulator-grounded debate to refine bodies and training rewards (Qiu & Cygan, 2025). PDA builds on the search-assistant role but grounds edits in module-level evidence and pairs them with testable behavioral predictions.

As trials accumulate, the choice of what to remember becomes part of the search. RoboMorph retains elite bodies and their rationales (Qiu et al., 2024), while Auto-Robotist grounds revisable structural skills in parent–child designs and their measured gains (Wang et al., 2026). PDA connects local behavioral evidence to an edit's predicted effect, then records prediction support separately from its performance gain. These two outcomes shape the next proposal: a useful body can be retained while its explanation is revised, whereas a realized behavioral change without a gain prompts reconsideration of the capability being targeted.

**Automated experimentation and research.** Using one experiment to guide the next extends beyond morphology design. Eureka generates and refines reward programs using reward-component training statistics (Ma et al., 2024), while MaestroMotif combines LLM-derived skill rewards with code-based skill composition (Klissarov et al., 2025). Reflexion reuses verbal feedback across attempts (Shinn et al., 2023); automated research and program-search systems likewise propose changes and evaluate them through execution (Lu et al., 2024; Novikov et al., 2025; Karpathy, 2026). PDA brings this iterative experimentation to robot co-design by choosing the next morphology ex-

periment within a fixed task and controller-learning protocol. Here, LLM-generated programs serve as measurement tools fixed after calibration, rather than reward functions. The loop investigates how structure shapes behavior within a finite training budget.

# 3 METHOD

PDA uses each completed experiment to guide the next (Figure 1). *Probe* trains and evaluates a body to produce an evidence report. *Diagnose* combines this report with past trials into a design brief, and *Adapt* proposes a body edit with a predicted metric change. The candidate returns to *Probe*, where performance guides selection and prediction checks inform subsequent reasoning. Algorithm 1 summarizes the procedure; Appendix G gives each LLM role's inputs and outputs.

## 3.1 PROBLEM SETTING AND INITIALIZATION

Let $\mathcal{M}$ denote the feasible morphology space and $m \in \mathcal{M}$ a robot body, including its structure and physical properties. We consider material-labeled voxel grids in *Evolution Gym* (Bhatia et al., 2021) and trees of rigid limbs connected by actuated joints in *UNIMAL* (Gupta et al., 2021). For each body, an MLP policy maps observations to actuator commands. Morphology therefore shapes both the robot's physical capabilities and the control problem to be learned.

For each body $m$, one training run from scratch uses $B$ environment steps and selects a controller $\pi_m = \mathcal{A}_B(m)$ under a fixed checkpoint rule. We freeze this controller and score the pair on a separate panel of $n$ episodes:

$$\widehat{J}_B(m) = \frac{1}{n}\sum_{i=1}^{n} J(m, \pi_m; \xi_i), \tag{1}$$

where $J$ is unnormalized episodic task return and $\xi_i$ specifies an evaluation seed. We cache $\widehat{J}_B(m)$ as the score of this trained pair. We report *search performance* as the highest score found among the pairs evaluated so far. The per-body budget $B$ fixes each candidate's learning opportunity, while $K$ limits additional trials after initialization. A run with $N_{\text{init}}$ initial bodies thus trains at most $N_{\text{init}}+K$ bodies, counting accepted and rejected candidates alike.

To initialize the search, the LLM uses task specifications, design constraints, and scene geometry to propose a small portfolio of bodies. After review, each body is trained and evaluated under the common protocol, and the highest-scoring pair supplies the starting design $m_0$. Rollouts of its controller support probe calibration (Section 3.2). The resulting behavioral measurements, panel score, and body–environment geometry form the initial report $x_0$. Each round $t$ then starts from the current design $m_t$, its cached controller, and report $x_t$.

## 3.2 PROBE: CONTROLLER LEARNING AND STRUCTURED EVIDENCE

**A shared measurement interface.** PDA first establishes a task-specific, *module-level measurement interface*. Given the task specification, initial morphology, and module-level observation/action definitions, the LLM specifies task-relevant factors through an executable transition interface $f(s, a, s')$, where $s$, $a$, and $s'$ denote the current state, action, and next state. Modules correspond to occupied voxels or named torso/limb bodies. Local factors are evaluated for each applicable module using shared definitions, while whole-body quantities, such as the robot's forward velocity, are explicitly labeled as shared. By preserving where motion or deformation occurs, this representation connects behavioral observations to components that can be edited.

The proposed probe is then calibrated during initialization. Evolution Gym implements the factors through generated code, whereas UNIMAL uses restricted arithmetic expressions over registered signals. Calibration checks whether the factors execute correctly on evaluation rollouts and reviews their definitions against the simulator's signal meanings, units, and observation scopes. In Evolution Gym, factors linked to known simulator quantities are recomputed using predefined formulas. Other generated factors retain an explicit label indicating that their physical interpretation has not been independently verified. After calibration, the retained measurement definitions are frozen throughout the search. Subsequent trials can therefore change the body and its learned behavior while using the

same definitions to describe what happens. Appendix D.1 details how reports preserve measurement identity and scope across body edits.

With the probe fixed, each candidate receives $B$ PPO training steps with a fresh MLP controller. Periodic evaluations select a checkpoint, whose controller is then frozen for the independent seven-episode selection panel. The panel supplies both the score in Eq. 1 and behavioral measurements for *Diagnose*, prediction checks, and memory. Evaluation interactions are recorded separately from PPO training. Appendices B and K give the Evolution Gym and UNIMAL settings.

To interpret these local measurements in the task context, PDA also records overall trajectory progress and body–environment geometry. Geometric descriptors are computed directly from the body specification and initial scene layout. For example, a climbing report pairs observed upward motion with the body's undeformed width and the channel width. When upward progress is limited, these measurements help the LLM assess whether to test a wider body.

**An evidence report with four complementary views.** *Probe* brings these measurements together in a structured evidence report. The *performance* view records the panel score alongside the learning curve, placing the achieved outcome in the context of controller learning. The *module factors* view summarizes local behavior over space and time, keeping observations tied to specific voxels or limbs. To place these local patterns in the task context, the *trajectory* view summarizes whole-body progress and the motion of task objects, while the *static fit* view describes body–environment geometry and candidate growth sites. The latter two views are supplied by deterministic platform adapters. Together, the four views allow *Diagnose* to relate task outcomes to both the behavior of individual components and the structure available for modification.

Each measurement retains its source ID, entity, units, and temporal scope so that its meaning remains explicit when reports are compared across trials. Fixed aggregation rules also record the number of contributing modules and the coverage of each measurement window. Quantities associated with absent module groups or unreached event windows are marked as unavailable. These annotations help distinguish changes in observed behavior from changes in what was available to measure (Appendix C.3). The measurements described in the report also provide the quantities that *Adapt* can use to predict and check the effects of a proposed edit.

### 3.3 Diagnose: From Evidence to a Design Brief

Building on the evidence report, *Diagnose* asks which capability the next body trial should investigate. Its inputs are the current report $x_t$ and context $\mathcal{W}_t$ from experimental memory, comprising a search summary and relevant past trial records. The LLM reads the learning curve alongside module-level behavior, trajectory progress, and body–environment geometry. The learning curve provides context for controller learning under the available budget, while behavioral and geometric observations suggest where a structural change may help. Together, these inputs guide a hypothesis about a possible body limitation, with controller coordination considered as an alternative explanation.

Experimental memory further helps refine this hypothesis through the outcomes of earlier interventions. Each retrieved record connects an exact body edit to its measured changes, task return, and prediction assessment. *Diagnose* uses these records to relate the current observations to what has already been tested and decide which uncertainty warrants another experiment. For example, a previous edit may have improved performance without producing its predicted behavioral effect, suggesting that the resulting body remains useful while its explanation needs revision. Section 3.5 describes how these records are constructed and retrieved.

The resulting *design brief* $h_t$ specifies a capability to test, the observations motivating it, and the uncertainty that the next trial should address. For instance, suppose a robot stalls at a terrain transition while deformation is concentrated near its rear. The brief might prioritize the ability to lift the front of the body over the transition, while noting that poor coordination of the existing actuators could also explain the stall. This gives the next trial a concrete behavioral objective. *Adapt* then determines which structural edit could realize that capability and what measurable change to predict.

### 3.4 ADAPT: FROM A DESIGN BRIEF TO A TRIAL PROPOSAL

Given the capability identified by *Diagnose*, *Adapt* determines how to investigate it through a concrete body change. The LLM receives the design brief $h_t$, a proposal base $b_t$—the body to edit, normally the current design $m_t$—and relevant trial records from experimental memory. Whereas *Diagnose* uses these records to refine the capability worth testing, *Adapt* examines earlier edits and their measured effects to formulate a new intervention. It proposes a candidate morphology $c_t$ and explains how the edit could produce the desired behavioral change. Edits may combine additions, removals, and changes to component properties.

To make this intervention testable, *Adapt* selects an available measurement from the evidence report and states how it should change. The prediction $\psi_t = (k_t, d_t, r_t)$ specifies the quantity $k_t$, expected direction $d_t$, and previously evaluated reference body $r_t$. Its stored measurement supplies the comparison baseline, retaining the same entity, units, and temporal scope. Thus, $b_t$ identifies the body being edited, while $r_t$ identifies the prediction baseline. Before training, the LLM explains the metric's connection to the proposed capability, and the system records its measurement role and any unresolved gap. In Evolution Gym, a separate LLM review checks this connection against the metric definition; programmatic checks bind the approved prediction to that definition. Appendix E.1 specifies these checks and each platform's reference. A candidate with an unresolved measurement gap can still proceed as a performance trial.

Before committing the training budget, the system computes the exact difference between $b_t$ and $c_t$, checks design validity and whether the body has already been evaluated, and compares the realized edit with its description. A bounded repair step addresses inconsistencies. The resulting proposal is saved in experimental memory before training, recording the candidate body, exact edit, proposed mechanism, and prediction or measurement gap. The candidate then returns to *Probe* to produce its evidence report. The following subsection describes how these results guide design selection, assess the prediction, and complete the trial record for subsequent decisions.

### 3.5 CLOSING THE LOOP WITH EXPERIMENTAL MEMORY

**Compare and select.** Once the candidate returns to *Probe*, its experiment yields an evidence report $x_t^c$ and selection score $\widehat{J}_B(c_t)$ for its trained body–controller pair. Compare & Select uses these results to assess performance and check the saved prediction separately. For performance, it compares the candidate with the cached trial of the current design through $\Delta_J = \widehat{J}_B(c_t) - \widehat{J}_B(m_t)$. Replacement requires this gain to meet a practical margin and exceed the comparison band from paired episode returns (Appendix E).

After evaluation, the program compares the candidate's value of $k_t$ with the stored value for $r_t$, checking measurement definitions and scopes. Fixed numerical rules assign supported, contradicted, or unclear status from the observed change. The saved measurement role and gap determine how this verdict informs the proposed capability (Appendix E.1). For example, higher vertical velocity without a return gain supports increased upward motion while leaving its task benefit unresolved.

Performance selection updates the current body and report together:

$$(m_{t+1}, x_{t+1}) = \begin{cases} (c_t, x_t^c), & \text{if the candidate is selected,} \\ (m_t, x_t), & \text{otherwise.} \end{cases} \tag{2}$$

The associated controller remains with this pair, and $x_{t+1}$ supplies the next *Diagnose* input. Meanwhile, every completed trial enters memory with its evidence and prediction assessment. A separate *best-observed record* retains the highest selection score, including improvements below the replacement margin. Thus, the current design provides a reference for continued search, while the best-observed record determines the final output.

**Memory for the next decision.** Experimental memory preserves the morphology search history by linking each candidate to the body from which it was edited and recording the exact changes between them. Both selected and rejected candidates remain in the archive $\mathcal{H}_t$. Each trial record contains the *proposal* (body edit, hypothesis, prediction or gap), *evidence* (return and measured changes), *assessment* (selection and prediction check), and *context* (budget and evaluation protocol). The proposal is saved before execution, and the evidence and assessment complete its record

afterward. Before each decision, memory supplies a compact summary of the online search history and at most three relevant completed trials:

$$\mathcal{W}_t = \big(G(\mathcal{H}_t),\ R_{\le 3}(\mathcal{H}_t; x_t)\big). \tag{3}$$

Both the summary $G$ and retrieval $R$ follow programmatic rules. The summary organizes previously tested edits and their outcomes, while retrieved records provide concrete examples of what those edits changed. Their construction accounts for the morphology representation on each platform, as detailed in Appendix F. *Diagnose* uses this history to refine the next capability to test, and *Adapt* draws on the relevant records to choose a concrete edit. In both stages, earlier explanations can be reconsidered alongside the measured behavior and performance.

**Continuing exploration and returning a result.** The current design $m_t$ normally supplies the proposal base $b_t$. To follow up on a promising intervention, PDA can also retain an unaccepted body for one further edit when it meets a non-inferiority criterion or has a supported functional prediction, under the bounded rules in Appendix E. The panel score is compared with that of the current design $m_t$, while the prediction check compares the corresponding measurements of $c_t$ and its declared reference body $r_t$. PDA returns the best-observed body–controller pair at the original per-body training budget.

# 4 Experiments

Our experiments ask whether PDA finds useful bodies efficiently, which observations and records support its revisions, and how the loop carries over to articulated bodies. Six Evolution Gym tasks provide the main performance comparison, with design traces showing how successive experiments inform edits. Controlled variants examine behavioral evidence, spatial localization, and experimental memory. Two UNIMAL tasks test the loop on articulated robots.

## 4.1 Experimental Setup

**Tasks and design space.** We use three locomotion tasks (Walker-v0, Climber-v0, and UpStepper-v0) and three manipulation tasks (Pusher-v0, Carrier-v0, and Thrower-v0) from Evolution Gym (Bhatia et al., 2021), all within a common $5 \times 5$ design space. Together, they test walking, vertical progress, terrain transitions, and different forms of object interaction. All language-guided methods receive the same public task/reward semantics and scene information, without prescribed successful structures. Appendix C details the task settings and observation interfaces.

**Baselines and shared settings.** We compare PDA with three bi-level optimization baselines from Evolution Gym (Bhatia et al., 2021): **GA**, **BO**, and **CPPN-NEAT**. Their outer loop searches over morphologies using a genetic algorithm, Bayesian optimization, or CPPN-based neuroevolution, respectively, while the inner loop trains a controller for each candidate morphology using PPO. We also evaluate two language-guided methods: **LASeR**, which uses diversity-oriented reflection (Song et al., 2025), and **Auto-Robotist**, which combines skill-conditioned proposals with evolutionary exploration and a revisable structural memory (Wang et al., 2026). Finally, **PDA-Initial** reports the mean score of PDA's selected initial pairs, providing a reference for subsequent search gains.

PDA and all five baselines share Evolution Gym's controller architecture and PPO settings (Bhatia et al., 2021), training a fresh controller for $B = 512{,}000$ steps per morphology. All use the same checkpoint-selection rule and independent seven-episode scoring panel. Language-guided methods use *DeepSeek-Flash*. PDA's budget comprises $N_{init} = 3$ initial bodies and $K = 50$ online candidates. Appendix B.2 gives the shared protocol and baseline implementations.

## 4.2 Search Performance under a Limited Budget

Figure 2 compares search performance against cumulative trained morphologies, including initialization for all methods. PDA uses 53 morphologies (three initial bodies and 50 online candidates), versus 300 or 500 for the baselines. Table 1 separates common-budget performance from longer-run endpoints. Efficiency here concerns morphology evaluations at fixed per-body training steps.[1]

[1] The supplementary website (`pda.html`) provides additional results and visualizations.

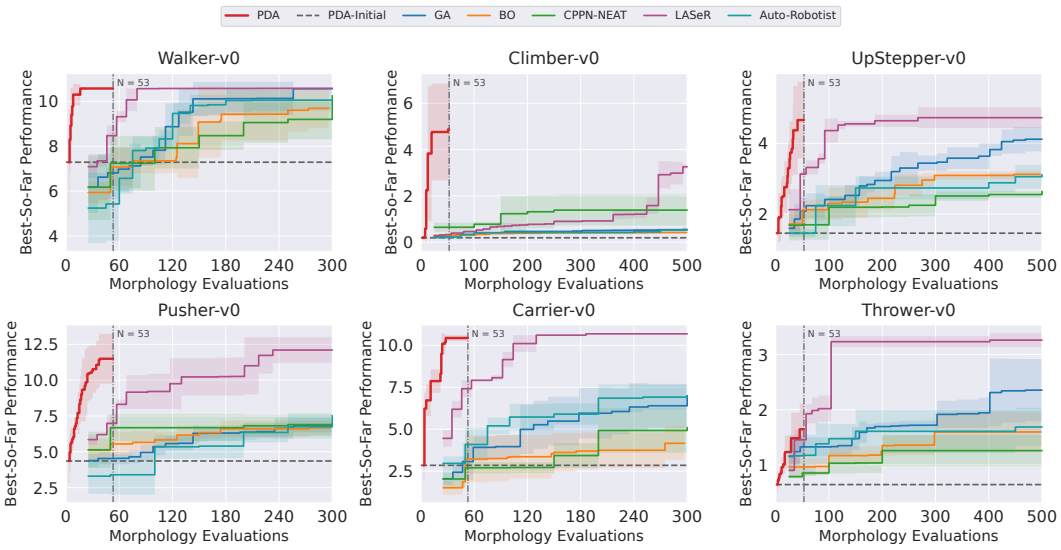


Figure 2: **Evolution Gym search performance.** Locomotion (top) and manipulation (bottom). Counts include initialization. Lines show means; shading denotes SD across six searches. Vertical dash-dot lines mark $N = 53$; horizontal dashed lines show PDA-Initial.

Table 1: **Common-budget and longer-run performance** (mean ± SD). Bold PDA means exceed the strongest baseline at $N = 53$. Baselines are ranked by mean in each column; parentheses give the method and endpoint budget.

| Task | PDA ($N = 53$) | Best baseline ($N = 53$) | Best baseline endpoint |
|---|---|---|---|
| Walker | **10.575** ± 0.004 | 8.474 ± 0.450 (LASeR) | 10.583 ± 0.014 (LASeR; 300) |
| Climber | **4.870** ± 2.061 | 0.645 ± 0.204 (CPPN-NEAT) | 3.257 ± 0.103 (LASeR; 500) |
| UpStepper | **4.663** ± 1.087 | 3.132 ± 0.067 (LASeR) | 4.731 ± 0.294 (LASeR; 500) |
| Pusher | **11.493** ± 1.740 | 6.988 ± 0.598 (LASeR) | 12.100 ± 0.900 (LASeR; 300) |
| Carrier | **10.439** ± 0.171 | 7.416 ± 0.500 (LASeR) | 10.688 ± 0.000 (LASeR; 300) |
| Thrower | **1.645** ± 0.464 | 1.453 ± 0.176 (LASeR) | 3.262 ± 0.127 (LASeR; 500) |

At 53 trained morphologies, PDA achieves the highest mean search performance on all six tasks. Longer baseline searches provide a second view: PDA reaches the leading Walker performance and exceeds all baseline endpoints on Climber. On UpStepper, Pusher, and Carrier, it exceeds the other baselines' final means and reaches levels that LASeR attains later; LASeR subsequently improves further. These gains reflect competitive designs early in search. On Thrower, PDA more than doubles its initial score and leads at $N = 53$, but LASeR and GA overtake it with longer search. Appendix J examines these trials.

### 4.3 Revising Designs and Explanations

The logged trials illustrate why design selection and prediction assessment play different roles. In Carrier run 5, trial 1 adds a rear wall and cuts a deck notch, raising the score from 3.247 to 6.564, although measured envelope overlap stays at zero. PDA retains the body but records the contradicted prediction. A central leg and soft foot in trial 2 reduce the score to 1.386. Recalling both outcomes, the LLM revises the overlap explanation and instead softens two spine cells in trial 3, reaching 7.192 with the predicted increase in object displacement. Both follow-ups edit trial 1's accepted body; Figure 4 in Appendix I.1 shows the sequence.

Thrower provides the converse case. In run 10, trial 16 increases post-peak object velocity from 0.619 to 3.591 as predicted, while the score falls from 1.699 to 0.853. This average starts at each trajectory's own speed peak, so a higher value can accompany a shorter throw. PDA rejects the can-

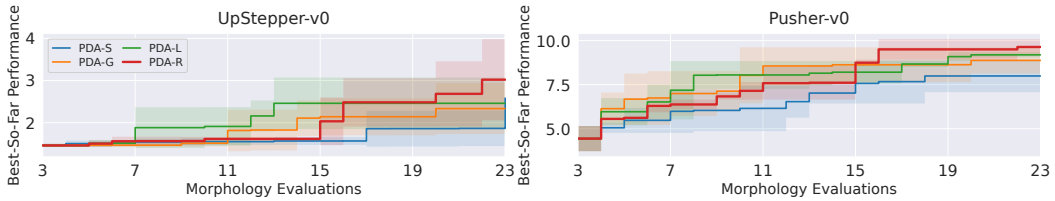


Figure 3: **Information ablations on two Evolution Gym tasks.** Lines show mean search performance; shading denotes SD across six seeds per task and variant.

didate but retains the supported metric prediction: an expected change occurred without improving performance (Appendix J).

### 4.4 Ablation Studies

Beyond the individual decisions in Section 4.3, we test search performance through four cumulative information variants: **PDA-S** supplies scores and learning curves; **PDA-G** adds global behavioral measurements and prediction targets; **PDA-L** restores module localization; and **PDA-R** adds hypotheses, predictions, measured changes, and assessments to history. All retain static geometry and body/edit/score history.

On UpStepper and Pusher, we use six seeds per task, sharing initialization across the four variants for each seed. Each search tests 20 online candidates under shared controller-learning, selection, and last-three-trials retrieval rules (Appendix I.2).

In Figure 3, behavioral measurements and prediction targets (PDA-S to PDA-G) raise the final mean on both tasks. Module localization (PDA-G to PDA-L) brings earlier gains on UpStepper; on Pusher, the curves cross before PDA-L finishes slightly ahead. Richer records (PDA-L to PDA-R) yield the highest final means after a slower start. These patterns support the complementary value of behavioral observations, their structural location, and accumulated experimental outcomes.

### 4.5 Beyond Voxel Bodies: UNIMAL

Finally, we apply PDA to *Incline* and *Push Box Incline*, using DERL's design space and MetaMorph's limb-level interface (Gupta et al., 2021; 2022). Platform adapters supply representations, measurements, and legal edits for the same Probe–Diagnose–Adapt loop. Each body receives approximately 5M training steps with a fresh MLP; policies and memory start anew. Figure 5 in Appendix K shows sustained gains on Incline and early improvements on Push Box Incline. These gains over initialization support applicability beyond voxel bodies.

## 5 Discussion and Conclusion

PDA turns controller-training runs into evidence for subsequent design. Fixed module-level measurements connect behavior to editable structure, while memory links exact edits, predictions, and outcomes. Separating performance from prediction support lets useful bodies be retained as explanations change. The six-task comparison, information ablations, and UNIMAL results support this approach to co-design under limited search budgets.

Thrower exposes a key limitation: a correctly predicted metric change may miss the capability needed for progress. Fixed definitions enable comparison but can overlook events important to task success. Retraining also couples morphology with learning variability, so prediction support concerns the resulting body–controller pair rather than an isolated structural cause. Physical deployment remains to be tested and adds fabrication constraints and simulator mismatch.

Future work could let unresolved hypotheses motivate new measurements, preserving earlier definitions or remeasuring references for comparison. Coupled with transferable trial memory, this would extend PDA toward embodied auto-research that chooses both interventions and the observations needed to assess them.

## AI USE STATEMENT

Large language models are part of PDA's experimental pipeline. They propose initial morphologies, generate probes that are calibrated and then fixed, interpret behavioral evidence, and propose body edits with predictions to be checked in subsequent trials. Controller training and simulator evaluation produce the performance scores and behavioral measurements. Section 3 and the accompanying appendices describe the models, prompts, and procedures used to validate and assess their outputs.

During research and manuscript preparation, we also used AI to process and visualize experimental logs, interpret results, and draft and revise the manuscript. These uses supported the research workflow alongside the LLM calls within PDA. The authors take responsibility for the final methodology, implementation, experimental results, interpretations, and presentation, including all AI-assisted content.

## REFERENCES

Jagdeep Bhatia, Holly Jackson, Yunsheng Tian, Jie Xu, and Wojciech Matusik. Evolution gym: A large-scale benchmark for evolving soft robots. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2201–2214, 2021.

Yifei Dong, Shaohang Han, Xianyi Cheng, Werner Friedl, Rafael I Cabral Muchacho, Máximo A Roa, Jana Tumova, and Florian T Pokorny. CageCoOpt: Enhancing manipulation robustness through caging-guided morphology and policy co-optimization. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 21795–21802. IEEE, 2025.

Jiawei Fang, Yuxuan Sun, Chengtian Ma, Qiuyu Lu, and Lining Yao. RoboMoRe: LLM-based robot co-design via joint optimization of morphology and reward. *arXiv preprint arXiv:2506.00276*, 2025.

George Jiayuan Gao, Tianyu Li, Junyao Shi, Yihan Li, Zizhe Zhang, Nadia Figueroa, and Dinesh Jayaraman. VLMgineer: Vision-language models as robotic toolsmiths. In *International Conference on Learning Representations*, volume 2026, pp. 91732–91763, 2026.

Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021.

Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. MetaMorph: Learning universal controllers with transformers. In *International Conference on Learning Representations*, 2022.

Kei Ikemura, Yifei Dong, and Florian T Pokorny. Latent diffeomorphic co-design of end-effectors for deformable and fragile object manipulation. In *Proceedings of Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.195.

Andrej Karpathy. Autoresearch: AI agents running research on single-GPU language-model training. https://github.com/karpathy/autoresearch, 2026.

Martin Klissarov, Mikael Henaff, Roberta Raileanu, Shagun Sodhani, Pascal Vincent, Amy Zhang, Pierre-Luc Bacon, Doina Precup, Marlos C. Machado, and Pierluca D'Oro. MaestroMotif: Skill design from artificial intelligence feedback. In *International Conference on Learning Representations*, volume 2025, pp. 18029–18057, 2025.

Guiliang Liu, Bo Yue, Yi Jin Kim, and Kui Jia. Embracing evolution: A call for body-control co-design in embodied humanoid robot. *arXiv preprint arXiv:2510.03081*, 2025.

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

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations*, volume 2024, pp. 26516–26560, 2024.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Kevin Qiu and Marek Cygan. Debate2Create: Robot co-design via multi-agent LLM debate. *arXiv preprint arXiv:2510.25850*, 2025.

Kevin Qiu, Władysław Pałucki, Krzysztof Ciebiera, Paweł Fijałkowski, Marek Cygan, and Łukasz Kuciński. RoboMorph: Evolving robot morphology using large language models. *arXiv preprint arXiv:2407.08626*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023.

Karl Sims. Evolving 3D morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.

Junru Song, Yang Yang, Huan Xiao, Wei Peng, Wen Yao, and Feifei Wang. LASeR: Towards diversified and generalizable robot design with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 58460–58489, 2025.

Nitish Sontakke, K Niranjan Kumar, and Sehoon Ha. RobotDesignGPT: Automated robot design synthesis using vision language models. *arXiv preprint arXiv:2601.11801*, 2026.

Luke Strgar and Sam Kriegman. Accelerated co-design of robots through morphological pretraining. In *International Conference on Learning Representations*, volume 2026, pp. 7450–7470, 2026.

Yunfei Wang, Xiaohao Xu, Yang Li, and Xiaonan Huang. When search becomes memory: Turning robot design trials into transferable skills. *arXiv preprint arXiv:2605.25832v1*, 2026.

Yuxing Wang, Shuang Wu, Haobo Fu, Qiang Fu, Tiantian Zhang, Yongzhe Chang, and Xueqian Wang. Curriculum-based co-design of morphology and control of voxel-based soft robots. In *The Eleventh International Conference on Learning Representations*, 2023a.

Yuxing Wang, Shuang Wu, Tiantian Zhang, Yongzhe Chang, Haobo Fu, Qiang Fu, and Xueqian Wang. PreCo: Enhancing generalization in co-design of modular soft robots via brain-body pre-training. In *Proceedings of the Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 478–498. PMLR, 2023b. URL https://proceedings.mlr.press/v229/wang23b.html.

Yuxing Wang, Zhiyu Chen, Tiantian Zhang, Qiyue Yin, Yongzhe Chang, Zhiheng Li, Liang Wang, and Xueqian Wang. Embodied co-design for rapidly evolving agents: Taxonomy, frontiers, and challenges. *arXiv preprint arXiv:2512.04770*, 2025.

Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris M. Kitani. Transform2Act: Learning a transform-and-control policy for efficient agent design. In *International Conference on Learning Representations*, 2022.

# A PDA PROCEDURE

Algorithm 1 summarizes Figure 1. The selected design supplies the next report, while every completed trial contributes to memory. The following sections specify the experimental settings, measurements, selection rules, and platform adapters.

**Algorithm 1** Probe–Diagnose–Adapt: the experimental loop

**Require:** Task and design constraints, LLM, learner $\mathcal{A}_B$, online candidate cap $K$, available search-training steps
**Ensure:** Best observed fixed-budget body–controller pair and experiment records

Generate and review initial bodies; train each with $\mathcal{A}_B$ and cache the results
Select $m_0$; calibrate and freeze measurements; construct its report $x_0$
Initialize the best-observed pair, online ledger $\mathcal{H}_0 = \emptyset$, and $t = 0$
**while** $t < K$ and a full trial fits within the remaining search budget **do**
  Retrieve global history and relevant trials as $\mathcal{W}_t$ (Eq. 3)
  **Diagnose:** $h_t \leftarrow \mathrm{Diagnose}(x_t, \mathcal{W}_t)$
  Choose proposal base $b_t$: the current body or an eligible retained body
  **Adapt:** $(c_t, \psi_t) \leftarrow \mathrm{Adapt}(h_t, b_t, x_t, \mathcal{W}_t)$
  Validate and save the candidate, exact edit, and prediction before training
  **Probe:** train a fresh controller for $c_t$ with $\mathcal{A}_B$; construct report $x_t^c$
  $a_t \leftarrow$ compare panel scores in $x_t^c$ and cached $x_t$ (Eq. 6)
  $v_t \leftarrow$ check prediction $\psi_t$ against measurements using its saved reference
  Select body $m_{t+1}$, its controller, and report $x_{t+1}$ using $a_t$ only
  $\mathcal{H}_{t+1} \leftarrow$ append proposal, evidence, both assessments, and context to $\mathcal{H}_t$
  Update the best-observed pair by selection score, independently of $a_t$
  $t \leftarrow t + 1$
**end while**
**return** Best-observed pair and all experiment records

Here Probe includes checkpoint selection and report construction. Return comparisons use the current design even when the proposal base differs; prediction checks use the saved reference. An invalid, untrained proposal does not advance $t$. Controller-only continuation uses a separate allocation and preserves the selected body's original $B$-budget result.

# B EXPERIMENTAL SETTINGS

We first give the shared budgets and learner settings, then describe the baseline protocols and curve construction. Task definitions and measurement inputs follow in Appendix C.

## B.1 BUDGETS AND CONTROLLER LEARNING

PDA and all five baselines share Evolution Gym's original controller architecture and PPO hyperparameters (Bhatia et al., 2021). Each body is trained from scratch for $B = 512{,}000$ steps, corresponding to 1000 PPO updates with four environments and 128 steps per environment. The network and PPO settings in Table 2 follow the benchmark's Table 6 and accompanying implementation; the table also specifies the checkpoint and selection-panel evaluations used in our comparison.[2] Table 3 separates morphology-search training from optional controller continuation.

**Checkpoint selection and candidate scoring.** During training, each scheduled evaluation runs one episode using the policy mean and the current observation-normalization statistics. A checkpoint replaces the saved controller when its unnormalized return exceeds the previous best, retaining the earlier checkpoint on an exact tie. After training, the selected controller and its observation normalizer are frozen and evaluated on an independent seven-episode selection panel. This panel uses

[2]The original implementation is available in Evolution Gym's PPO code; its linked policy implementation supplies the network defaults.

Table 2: Shared Evolution Gym controller settings for PDA, GA, BO, CPPN-NEAT, LASeR, and Auto-Robotist.

| Parameter | Value |
|---|---|
| Actor / critic | Separate 64–64 MLPs, Tanh |
| Policy | Diagonal Gaussian; learned log std. |
| Initial log standard deviation | 0 |
| Optimizer / learning rate / Adam $\epsilon$ | Adam / $2.5 \times 10^{-4}$ / $10^{-5}$ |
| Learning-rate schedule | Linear decay |
| GAE / discount $\gamma$ / GAE $\lambda$ | Enabled / 0.99 / 0.95 |
| Parallel environments / rollout length | 4 / 128 |
| PPO epochs / minibatches per update | 4 / 4 |
| PPO clip / value loss / entropy coefficient | 0.1 / 0.5 / 0.01 |
| Value clipping / maximum gradient norm | Enabled / 0.5 |
| Proper time-limit handling | Disabled (original PPO default) |
| Observation / training-reward normalization | Enabled |
| Updates / total training steps per body | 1000 / 512,000 |
| Checkpoint evaluation interval | 50 updates (25,600 training steps) |
| Episodes per checkpoint evaluation | 1 |
| Selection panel / action choice | 7 episodes / deterministic policy mean |

fixed environment seeds separate from the checkpoint evaluations. Its mean unnormalized return defines the trained pair's score $\widehat{J}_B(m)$ in Eq. 1. These scores guide the adaptive search and provide the measurements reported in the search-performance curves and design traces. PDA and all five baselines use the same checkpoint-selection rule, selection-panel configuration, and scoring procedure.

**Behavioral evidence from the selection panel.** For each online PDA candidate, the selection-panel trajectories also supply the module-level measurements and trajectory summaries in its evidence report. The same evaluation therefore produces both the mean return for body selection and the observations used by Diagnose, prediction checks, and memory. Aggregated factors retain their contributing episode and module counts, while representative-episode quantities retain that narrower scope. At initialization, probe calibration and the subsequent evidence refresh require additional evaluations of the selected initial controller.

**Accounting for environment interactions.** The per-body budget $B$ counts transitions used for PPO training. Checkpoint evaluations, independent selection-panel evaluations, and initialization probe calibration and evidence refresh contribute additional environment interactions, whose steps are recorded separately from training. In online trials, scoring and behavioral measurement share the selection-panel trajectories, so their interaction cost is counted once. Table 3 reports training allocations; the curves count trained morphologies, including all initial bodies and accepted or rejected candidates.

Table 3: Configured full-run training budgets. Continuation is separate from the fixed-budget search results. UNIMAL curves use a 40-evaluation reporting horizon.

| Allocation | Evolution Gym | UNIMAL |
|---|---|---|
| Per-body steps, $B$ | 512,000 | 4,997,120 |
| Initial bodies | 3 | 3 |
| Online candidate cap, $K$ | 50 | 50 |
| Search-training cap, $53B$ | 27,136,000 | 264,847,360 |
| Final continuation allowance, $6B$ | 3,072,000 | 29,982,720 |
| Total configured training steps, $59B$ | 30,208,000 | 294,830,080 |

The information ablations use three initial bodies and 20 online trials: $23B = 11{,}776{,}000$ training steps per search, without controller continuation. Each variant uses six seeds per task; the four

variants share initialization for each task and seed (Appendix I.2). The same separation of training and evaluation interactions applies. Each search is charged the shared initialization once; rereading a cached result does not count as another trained body.

### B.2 Baseline Protocols

GA, BO, and CPPN-NEAT use voxel mutation, surrogate-based optimization, and evolving spatial encodings, respectively. LASeR retains its proposal and diversity-reflection mechanisms (Song et al., 2025). All methods use the same six tasks, a $5 \times 5$ design envelope, and 512K training steps per body. All language-guided methods receive the same public task, reward, and scene information and use DeepSeek-Flash. PDA-Initial is the mean selected-initial score across PDA runs, with no subsequent search.

**Auto-Robotist.** We reproduce the authors' cold-start $5 \times 5$ setting (Wang et al., 2026), with an empty skill library for each task and seed. Its structural archetypes, positive/negative rules, GA exploration, and Add/Diagnose/Merge updates are retained. Each generation has 25 proposal slots: 15 LLM-conditioned and 10 GA proposals. DeepSeek-Flash is used in all LLM roles at temperature 1.0. This evaluates cold-start search rather than cross-task skill transfer.

**Shared training and evaluation.** Table 4 summarizes the shared training and scoring protocol across the six search methods. Controller architecture, PPO hyperparameters, checkpoint selection, and panel scoring are fixed throughout the comparison. The methods differ in how they propose bodies and use the resulting evaluations to guide search.

Table 4: Cross-method protocol in Evolution Gym. All methods share the learner and checkpoint-selection settings in Table 2. Each candidate is scored by the mean return on an independent seven-episode panel with its selected controller frozen.

| Method | Training steps | Learner / checkpoint | Candidate score | Count initialization |
|---|---|---|---|---|
| PDA | 512K | Shared | 7-episode mean | Yes |
| GA | 512K | Shared | 7-episode mean | Yes |
| BO | 512K | Shared | 7-episode mean | Yes |
| CPPN-NEAT | 512K | Shared | 7-episode mean | Yes |
| LASeR | 512K | Shared | 7-episode mean | Yes |
| Auto-Robotist | 512K | Shared | 7-episode mean | Yes |

**Curve construction.** For search run $r$, let $m_{r,k}$ be its $k$th trained body and $\widehat{J}_{B,r}(m_{r,k})$ the cached panel score of that body and its selected controller. Search performance after $N$ trained morphologies is

$$S_r(N) = \max_{1 \le k \le N} \widehat{J}_{B,r}(m_{r,k}). \tag{4}$$

Initialization contributes to $N$ for every method. PDA's three initial bodies occupy evaluations 1–3, and online trial $t$ is plotted at evaluation $3 + t$. Each accepted or rejected trained candidate adds one evaluation; rereading a cached result adds none. Curves show the mean and standard deviation of $S_r(N)$ across six independent searches. Each search trains one controller per candidate, so variation across runs reflects the combined effects of morphology proposals and controller learning. The seven panel episodes characterize each fixed pair within a run. PDA-Initial is the mean best score from the three-body initialization, held as a horizontal reference.

**LLM settings.** DeepSeek-Flash is accessed through an OpenAI-compatible Chat Completions API with the model field `deepseek-flash`. This is the recorded service identifier. In Evolution Gym, PDA uses temperature 0.4 for initial body generation, online Adapt, and proposal repair, and 0.0 for Diagnose and review, including the pre-trial metric-alignment review. UNIMAL task configurations use temperature 0.2. The recorded Evolution Gym runtime uses Python 3.7.11, PyTorch 1.11.0 with CUDA 11.3, NumPy 1.21.5, and Gymnasium 0.28.1. Appendix G illustrates the role-specific prompts; exported run configurations retain API model and sampling settings alongside runtime versions.

The efficiency comparison uses morphology evaluations with fixed per-body controller training. Evaluation interactions are accounted for as described above; LLM inference and simulator throughput additionally affect elapsed time. The reported morphology budgets therefore quantify search progress rather than wall-clock speedup.

# C Evolution Gym Tasks and Evidence

## C.1 Tasks and Design Space

We use the six Evolution Gym tasks in Table 5 (Bhatia et al., 2021). Each body occupies a $5 \times 5$ canvas with five material labels: empty, rigid, soft, horizontal actuator, and vertical actuator (0–4). Occupied cells must form a connected body with at least one actuator. The task rewards and termination rules remain unchanged.

Table 5: Evolution Gym tasks and maximum episode lengths.

| Task | Horizon | Objective and scene |
|---|---|---|
| Walker-v0 | 500 | Move forward on level ground. |
| Climber-v0 | 400 | Move upward inside a vertical channel. |
| UpStepper-v0 | 600 | Move forward over an ascending surface. |
| Pusher-v0 | 500 | Push a soft payload initially ahead of the robot. |
| Carrier-v0 | 500 | Move forward while carrying a soft payload initially above the body. |
| Thrower-v0 | 300 | Propel a soft payload forward across fixed obstacles. |

World axes point rightward ($+x$) and upward ($+y$); canvas rows increase downward and columns rightward. An undeformed voxel spans 0.1 world units. Terrain and body arrays have separate origins, and external-object placement follows the task definition. Task prompts provide these conventions, scene geometry, object placement, and reward/termination conditions without prescribing a body or gait.

## C.2 Controller and Measurement Inputs

The MLP receives the environment's original observation vector. The measurement interface separately exposes unnormalized module signals, summarized in Table 6. This separation lets probes refer to editable cells without changing the controller's input representation.

Table 6: Evolution Gym module observations. Slices are zero-based and half-open; $d_g$ is the task-global observation dimension. Occupancy and actuator masks are supplied separately.

| Slice | Meaning |
|---|---|
| $[0, 2)$ | Canvas row/10 and column/10. |
| $[2, 4)$ | Module-center world velocity $[v_x, v_y]$. |
| $[4, 12)$ | Four corner positions relative to robot COM, ordered as four $x$ then four $y$ coordinates. |
| $[12, 12 + d_g)$ | Task-global observations shared across occupied cells. |
| $[12 + d_g, 17 + d_g)$ | One-hot material: empty, rigid, soft, horizontal or vertical actuator. |

Angles are in radians, and object-relative position is object COM minus robot COM. Floor and ceiling sensors measure vertical distances from robot COM, clipped to $[0, 0.5]$ world units and sampled at horizontal cell offsets. These distances and geometric overlap describe spatial relations rather than contact forces. New growth sites come from static geometry because empty cells have no measured activity.

The controller uses Evolution Gym's original Gaussian action interface. For a sampled policy action $u$, the environment applies the rest-length multiplier $a = \mathrm{clip}(1 + u, 0.6, 1.6)$ to each active cell along its actuation axis. PPO computes likelihoods for the sampled Gaussian actions, and evaluation

uses the policy mean with the same environment transformation. Full task observation layouts and reward definitions accompany the released configurations.

### C.3 Evidence Construction

The four report views in Figure 1 connect a trained pair's performance to its behavior and editable structure (Table 7). The LLM proposes measurement definitions and design hypotheses; simulator rollouts and deterministic evaluators supply their numerical values.

Table 7: The four evidence-report views and their roles.

| View | Measurements | Use in reasoning |
|---|---|---|
| Performance | Selection-panel score, checkpoint evaluation curve, saved controller | Assess improvement and controller adaptation. |
| Module factors | Transition measurements summarized by module and time window | Locate motion or deformation in editable structure. |
| Trajectory | Task-axis motion, relative geometry, temporal summaries | Identify relevant stages of task execution. |
| Static fit | Body–scene geometry and legal growth sites | Relate the proposed change to physical constraints. |

**Generated and fixed components.** Platform adapters supply task definitions, observation layouts, trajectory summaries, measurement primitives, and aggregation rules. Within this interface, the LLM selects static measurement intents and defines dynamic factors relevant to the task. Evolution Gym recomputes reference-backed factors with canonical formulas, while custom factors retain their interpretation status. Together, these measurements connect task-level progress to local behavior and body geometry. Their definitions remain fixed as the loop compares edits and accumulates experimental records; the task reward continues to determine performance.

**Aggregation across modules and time.** Probe outputs are masked by occupancy and factor applicability. Shared quantities appear once, while local factors retain summary statistics and selected low/high module references. Temporal summaries use declared windows, including terminal observations before reset. Each quantity carries its source ID, entity, units, aggregation, and coverage. These annotations allow a cited observation to be traced back to the part of the rollout it describes.

**Comparing changing bodies.** Measurement definitions remain fixed, but the measured population can change. Removing a voxel makes its response unavailable; changing its material can change factor applicability. Dynamic regions, such as the leading edge, are resolved on each body. Population counts and window coverage therefore accompany aggregates. An absent material class or an unreached event window has no comparable mean, whereas a valid constant signal remains an observation.

Numerical validation, semantic review, and prediction suitability serve different purposes: they check execution, the physical meaning of a quantity, and its relevance to the proposed hypothesis. For example, actuator count describes capacity, while realized motion describes behavior. Keeping these roles explicit helps later trials revise an explanation when a measured change and task return disagree.

## D Probe Calibration and Consistency

**Establishing the interface.** Initialization starts with static measurement intents derived from the task and design space. Supported geometric computations inform three body proposals with different intended mechanisms. After blinded review and at most one revision, each body receives $512{,}000$ training steps. The highest mean return on the independent selection panel chooses the initial design. A module probe is generated in the reviewed-body context; if another body wins, it is regenerated once for that body before calibration.

**Calibrating and freezing.** Source and runtime checks test output shape, finite values, applicability masks, and bounded computation. Rollouts from the selected controller then support a semantic review and at most one repair. Usable factors are frozen; excluded factors remain unavailable. Raw values and fixed linear scales are supported, while nonlinear transforms require declared scales and ranges. Thus, subsequent trials change the body and observed values while retaining the same measurement definitions and task reward. Appendix G gives the corresponding prompts.

### D.1 Measurement Scope across Body Edits

Each run retains its calibrated factor definitions throughout search. Reports carry the source, entity, coordinate frame, units, and aggregation of each measurement, together with the contributing module and episode counts. These annotations connect a value to the observations it summarizes and provide the comparison context for later prediction checks.

As bodies change, occupancy and applicability masks determine which modules contribute to each factor. Local measurements retain their module identities, while whole-body quantities retain their shared scope. Absent material classes and unreached event windows produce unavailable measurements. Candidate and reference values enter a prediction check when their definitions and scopes are compatible; an unavailable comparison remains recorded in the trial. This preserves a common measurement vocabulary while making changes in the observed population explicit.

## E Trial Comparison and Prediction Checks

Each trial uses the platform's fixed learner and scoring protocol (Appendices B and K). The current body–controller pair's cached $B$-budget record provides the reference score for the next comparison.

**Selection from panel scores.** Let $p$ denote the current design and $c$ the candidate. Their score difference is $\Delta_J = \widehat{J}_B(c) - \widehat{J}_B(p)$, with reference score $\bar{J}_p = \widehat{J}_B(p)$. The practical improvement margin is

$$\epsilon(p) = \max\big(\epsilon_{\text{abs}}, \epsilon_{\text{rel}}|\bar{J}_p|\big)\,. \tag{5}$$

The candidate and current design are evaluated on the same fixed panel of $n = 7$ episodes, with the current design's returns read from its cached record. Define $q = z\, s_\Delta/\sqrt{n}$, where $s_\Delta$ is the sample standard deviation of the paired return differences. The decision rule is

$$V_J(c,p) = \begin{cases} \textsc{Improved}, & \Delta_J \geq \epsilon(p) \text{ and } \Delta_J > q, \\ \textsc{Degraded}, & \Delta_J \leq -\epsilon(p) \text{ and } \Delta_J < -q, \\ \textsc{Unresolved}, & \text{otherwise.} \end{cases} \tag{6}$$

Both platforms use $\epsilon_{\text{abs}} = 0.001$, $\epsilon_{\text{rel}} = 0.01$, and $z = 1$. The band $q$ scales the online replacement threshold with episode-level variation for the two fixed controllers. When paired return differences are identical, $q = 0$ and the practical margin determines selection. Training variability enters through the controller learned for each candidate; the reported between-search standard deviation reflects this variability together with differences in the search trajectory. Learning-curve summaries provide context for diagnosis.

An Improved score selects the candidate; the other two outcomes retain the current body. Selection replaces the body, controller, and report together. These performance assessments govern the next search reference, while prediction assessments guide interpretation of the edit. Separately, the best-observed archive retains the highest selection score, including gains below the replacement margin. Cached records preserve the exact body, controller, normalizer, and evaluation protocol. Controller continuation uses a separate record, preserving the fixed-budget search scores.

### E.1 Prediction Definition and Assessment

A pre-trial prediction $\psi_t = (k_t, d_t, r_t)$ specifies a measured quantity, direction, and reference. The quantity retains its entity, coordinate frame, time window, aggregation, and units. Evolution Gym compares both return and predicted changes against the current design, including for follow-up edits. UNIMAL compares predicted changes against the actual proposal base while retaining the current design as the return reference. Records preserve both bodies so that the tested contrast is explicit.

**Who checks the prediction, and when.** In Evolution Gym, Adapt chooses a target capability and metric and explains their connection. Before training, the program checks metric availability and assigns its role from the measurement catalog: a metric of the target capability stage or a checked auxiliary observation. A separate call to the same LLM then reviews the proposed claim against the metric's definition, entity, statistic, and time scope, using the proposal and existing evidence. The program requires a passing review with a reason and matching metric identifier and statistic type. An unresolved or unavailable review converts the prediction to a measurement gap while preserving the feasible body trial. The resulting definition, role, review, and reference are saved before candidate evaluation. Afterward, fixed programmatic rules compute the numerical verdict and functional-consistency assessment from those saved fields and the measured values.

**Evolution Gym continuous quantities.** Let $a = k_t(r_t)$, $b = k_t(c_t)$, $\delta = b - a$, and $s = \max(|a|, |b|)$. An increase has $d_t = +1$ and a decrease $d_t = -1$. The directional check is

$$V_\psi(c_t, r_t) = \begin{cases} \textsc{Supported}, & d_t\delta > \tau, \\ \textsc{Contradicted}, & d_t\delta < -\tau, \\ \textsc{Unclear}, & |\delta| \le \tau \text{ or measurement unavailable.} \end{cases} \tag{7}$$

For reference-checked raw or linearly scaled Evolution Gym factors, $\tau = \max(0.02s, 32\epsilon_{\text{mach}}s)$, which preserves the verdict under a common positive unit conversion. Other continuous metrics use $\tau = \max(10^{-4}, 0.02\max(s, 10^{-4}))$ in their declared units. These tolerances set the detectable directional change in observations of the two trained pairs. Checks use the stored precision and retain panel or representative-episode scope, tying each assessment to the behavior measured in that trial.

**Events and missing values.** For a directly observed binary event, failure to change in the predicted direction contradicts the prediction. A panel-average event frequency instead uses the continuous rule. Missing conditional quantities remain unavailable: an absent event has no onset time, and an unreached window or absent material class has no within-window or population mean. A small observed change and a missing comparison are therefore recorded separately, although both can yield an unclear assessment.

**From observations to hypotheses.** For a matched Evolution Gym capability-stage metric, the functional-consistency assessment inherits the numerical verdict. For an auxiliary observation, the numerical verdict is retained but functional consistency remains unclear. A measurement gap or incompatible comparison likewise leaves the function unresolved. Thus, metric roles are determined before the trial, and the post-trial assessment applies fixed rules rather than asking the LLM to reinterpret the outcome. Table 8 separates the resulting records. Supported functional consistency denotes agreement with the declared test under its measurement scope. Auxiliary agreement can inform later reasoning but does not qualify a functional follow-up allowance. Body selection continues to follow Eq. 6.

**UNIMAL adapter.** Diagnose and Adapt specify the physical metric, direction, optional target, and any measurement gap before execution. The program verifies metric availability and carries forward gaps identified by Diagnose. After evaluation, it compares the candidate with its proposal base using $\tau = \max(10^{-8}, 0.03|a|)$. A directional change of at least $\tau$ supports the observation prediction if any declared target is also reached; an opposite change of at least $\tau$ contradicts it. Otherwise, or when measurements are unavailable, the verdict is unclear. A saved measurement gap keeps the functional assessment unclear even if the observable prediction is supported. Episode length, energy, control cost, and action magnitude are marked descriptive-only. These adapter rules retain the same separation between measured changes and their interpretation.

**One-step follow-up exploration.** An unaccepted online body can supply one follow-up base if its performance assessment is unresolved with non-inferior terminal and learning-curve evidence and no contradicted prediction, or if it has a supported functional-consistency assessment. The latter allowance is used at most once per current design. A follow-up child cannot start another chain. Its record retains both the edit from its proposal base and the difference from the current design used for score comparison.

Table 8: Three distinct uses of experimental results. Performance guides selection; prediction assessments interpret the observed change and its relevance to the proposed capability.

| Assessment | Question | Consequence |
|---|---|---|
| Performance assessment $V_J$ | Does the panel score meet the improvement margin and exceed the comparison band? | Improved selects the candidate; degraded or unresolved retains the current design. |
| Metric verdict $V_\psi$ | Does the declared quantity change in the predicted direction? | Supported, contradicted, or unclear; records numerical agreement. |
| Functional consistency | Does that measurement test the proposed capability? | Interprets the metric verdict with its role; auxiliary agreement alone leaves the function unresolved. |

**Search output and controller continuation.** PDA returns the best-observed body–controller pair from the search archive. Any final controller-only training holds that body fixed and uses a separate allocation; the search curves report performance at the original per-body budget.

## F Experimental Memory

Memory connects each proposed change to its measured result. A record is opened before training and completed afterward, so later reasoning can distinguish a prediction from an observation. Each search starts with an empty online archive; initial portfolio results remain available as separate body comparisons.

Table 9: Information retained for each completed trial.

| Component | Contents |
|---|---|
| Proposal | Reference and candidate bodies, exact edits, hypothesis, predicted metric change or measurement gap. |
| Evidence | Selection-panel scores, learning curves, and measured changes with source IDs and scopes. |
| Assessment | Separate body-selection and prediction outcomes, including unsuccessful trials. |
| Context | Task stage, learning budget, evaluation protocol, and comparison reference. |

**From the archive to a decision.** In Evolution Gym, the summary groups completed trials by capability stage and structural edit pattern. Retrieval considers the current task stage, observed failure, and body geometry. Priority goes to an active follow-up base and the trial that produced the current design, followed by relevant failures or near misses. A candidate-specific realization check can also retrieve a similar failed intervention.

In UNIMAL, the summary combines the morphology search graph, outcomes grouped by edit operation, and a compact index of tested interventions. Retrieval includes the best accepted trial when available, a high-performing unresolved trial, and unsuccessful trials, using a shared parent body, similarity in limb count and degrees of freedom, and recency to rank relevant records. Both platforms supply at most three distinct detailed trial records alongside the summarized history.

The selected cases retain exact bodies, edits, measured changes, and both assessments. Outcome-derived summaries keep their interpretation tied to the tested intervention. For example, an improved body with a contradicted prediction remains a useful design but calls for a revised explanation. Conversely, a supported prediction without a return gain records an observed change whose task benefit remains unresolved.

**Storage and information limits.** The full archive remains on disk, while LLM inputs contain the summary and retrieved cases. Evolution Gym targets 96 KiB per decision packet with a 256 KiB hard limit; repeated prose is reduced before exact edits and outcomes. UNIMAL uses a 512 KiB limit. These are serialized byte limits, not token counts. Thus, disk storage grows with the number of trials while each decision receives bounded context. The PDA-R versus PDA-L ablation tests richer record content under a shared last-three-trials retrieval rule (Appendix I.2).

# G Prompt Templates

These condensed templates show the inputs and outputs of each LLM role. Angle brackets denote runtime content, and ellipses omit secondary fields. We give Evolution Gym examples followed by the UNIMAL substitutions. Full prompts accompany the code.

## G.1 Shared Context and Initialization

Calls receive public task/reward semantics, scene geometry, the legal body space, observation/action meanings, and the fixed training budget. Initialization uses separate requests for measurement planning, body proposals, and review.

```
SHARED CONTEXT
Task and scene: <objective, reward, terrain, object placement>
Body space: <5x5 canvas, materials 0..4, validity constraints>
Signals: <observation/action fields, frames, units, array layout>
Learning: <fresh MLP/PPO, fixed steps per body>

INITIALIZATION CALLS
Static plan: <task + measurement registry> -> metric intents
Body proposals: <task + static plan> -> bodies and mechanisms
Blinded review: <bodies + static measurements> -> assessments
Train reviewed bodies; select the initial design by return.
```

## G.2 Probe

Probe generation receives the initial body and static plan. It requests interpretable transition factors whose definitions can be reused as the body changes.

```
ROLE: Define 4-6 interpretable transition factors.
INPUT: <shared context, initial body, static plan>
Use module_state, module_action, module_next_state.
Preserve physical units and distinguish local from shared signals.
Declare applicability; prefer raw values or fixed linear scaling.
OUTPUT (selected fields):
{
  "factors": [{
    "factor_name": "<name>",
    "explanation": "<physical meaning>",
    "spatial_scope": "module_local | global_shared",
    "applicable_module_types": [1, 2, 3, 4],
    "measurement_contract": {"reference": "<quantity>", ...}
  }, ...],
  "function": "def probe_func(...): ... return probe_factors"
}
Return a finite, vectorized (batch, number_of_factors) array.
```

Calibration review checks definitions and code against rollout statistics and signal conventions. At most one repair precedes freezing. Subsequent values are computed by the evaluator, with the reward unchanged.

### G.3 DIAGNOSE

Diagnose combines the selected-design report with a global search summary and up to three retrieved trials. Evidence IDs identify observations; metric IDs identify quantities available for prediction.

```
ROLE: Identify a capability to test before choosing a body edit.
INPUT: <current body and evidence report, metric catalog,
        search summary, retrieved trial cards>
Compare a possible body limitation with incomplete control learning.
Use past edits and measured outcomes to refine the next question.
Keep historical explanations distinct from observations.
OUTPUT (selected fields):
{
  "decision": "MODIFY",
  "controller_opportunity": "<alternative learning explanation>",
  "morphology_learnability_bottleneck": "<testable hypothesis>",
  "decision_rationale": "<evidence-based comparison>",
  "evidence_ids": ["<visible IDs>"],
  "experience_ids": ["<retrieved IDs; empty if unavailable>"],
  "experience_assessment": "<judgment update and next test>",
  "uncertainty": "<main ambiguity>",
  "design_objective": "<physical capability to test>"
}
```

### G.4 ADAPT

Adapt receives the brief, exact proposal base, cited evidence, relevant history, legal edits, and available measurements. It proposes one intervention with a predicted metric change, rather than a numerical fitness estimate.

```
ROLE: Turn the brief into one feasible, untested body change.
INPUT: <diagnosis, proposal base, cited observations, trial cards,
        metric catalog, legal edits, previously tested bodies>
State the hypothesis and complete edit set.
Choose a measurement with matching entity and time scope.
Use MEASUREMENT_GAP when no available metric tests the hypothesis.
OUTPUT (selected fields):
{
  "hypothesis": "<what this intervention tests>",
  "mechanism": "<proposed physical pathway>",
  "experience_rationale": "<relevant historical evidence>",
  "evidence_ids": [...], "experience_ids": [...],
  "capability_test": {
    "status": "MEASURED | MEASUREMENT_GAP",
    "metric_id": "<catalog ID, or NONE>",
    "direction": "increase | decrease | NONE",
    "rationale": "<measurement relevance or missing quantity>", ...
  },
  "edits": [{"position": [<row>, <column>],
             "new_type": <integer 0..4>, "rationale": "..."}], ...
}
Save the proposal before training.
```

An intervention may edit several cells. Validation checks the realized body and novelty, while review compares the exact edit with its stated mechanism. A mismatch permits one repair and re-audit. The trial then measures outcomes and selects bodies by return, independently of prediction agreement.

### G.5 UNIMAL SUBSTITUTIONS

UNIMAL uses the same roles with limb/joint observations and DERL's design operations. Probes are restricted arithmetic expressions rather than Python functions; Adapt returns an operation on a symmetry group.

```
PROBE INPUT: <task, registered metrics, raw module signals>
PROBE OUTPUT:
{"factors": [{"name": "<name>", "expression": "abs(vx)",
  "aggregation": "all_bodies | limbs_only",
  "meaning": "<quantity, units, population>", ...}], ...}

ADAPT INPUT: <brief, body, evidence, history, legal operations>
ADAPT OUTPUT:
{"operation": "<typed mutation>", "target_groups": [<group IDs>],
 "parameter_values": {"<parameter>": <legal value>},
 "observable_prediction": "<expected physical change>",
 "prediction_metric": "<metric ID>",
 "prediction_direction": "increase | decrease",
 "prediction_target": null, "measurement_gap": "", ...}
```

Here `abs(vx)` illustrates syntax rather than a prescribed factor. Group IDs identify editable symmetry groups. An optional numerical target is supported, but a prediction need not estimate task return. Appendix K describes the platform-specific inputs and operations.

## H A LOGGED PROBE AND EVIDENCE REPORT

The following example comes from Walker-v0, seed 2, after initialization and before the first online proposal. The generated function and evidence values are taken from the same saved run.

### H.1 GENERATED PROBE

Listing 1 shows the complete function. The helper `probe_measure` reads named quantities from raw module observations. Its five outputs are module forward velocity, module vertical velocity, area increment, actuator rest-length offset, and shared body forward velocity.

Listing 1: Logged Walker-v0 probe. NumPy and the measurement helper are supplied by the evaluator.

```
def probe_func(module_state, module_action, module_next_state):
    ms = np.asarray(module_state, dtype=float)
    mn = np.asarray(module_next_state, dtype=float)
    vx0 = probe_measure(module_state, module_action, 'voxel_vel', 0)
    vx1 = probe_measure(module_next_state, module_action, 'voxel_vel',
    0)
    vy0 = probe_measure(module_state, module_action, 'voxel_vel', 1)
    vy1 = probe_measure(module_next_state, module_action, 'voxel_vel',
    1)
    a0 = probe_measure(module_state, module_action, 'voxel_area')
    a1 = probe_measure(module_next_state, module_action, 'voxel_area')
    act = probe_measure(module_state, module_action, 'actuator_offset')
    body_vx = 0.5 * (ms[:, 12] + mn[:, 12])
    f_vx = 0.5 * (vx0 + vx1)
    f_vy = 0.5 * (vy0 + vy1)
    f_area = a1 - a0
    out = np.stack([f_vx, f_vy, f_area, act, body_vx], axis=1)
    return np.nan_to_num(out, nan=0.0, posinf=0.0, neginf=0.0)
```

The velocity factors average transition endpoints. Area increment is the difference between successive areas, while actuator offset is measured from the neutral multiplier 1.0 and applies only to

actuator cells. The final factor reads a shared velocity field at column 12 of this run's observation layout. Its custom interpretation retains an unverified label. Occupancy and applicability masks are applied externally before aggregation.

The final `nan_to_num` is preserved from the archived function. It can replace non-finite intermediate results with zero before the outer finite-output check. Reference-backed factors are subsequently recomputed by canonical formulas, whereas custom factors retain this possible ambiguity. Occupancy masks do not establish numerical validity: auditing a sanitized custom output requires the original inputs and unsanitized computation, not the stored finite value alone.

### H.2 Evidence Report

Listing 2 preserves selected fields and values from the report supplied to Diagnose. It includes all four report views, with one of five factors and two evidence IDs shown for brevity. The full report also contains the body, learning budget, other factors, temporal summaries, and history.

Listing 2: Excerpt of the logged Walker evidence report.


```
{
  "task_name": "Walker-v0",
  "measurement_status": {"status": "COMMISSIONED", "frozen": true},
  "evidence_cards": {
    "outcome": {"current_reward_mean": 4.04313},
    "controller_learning": {
      "evaluation_role": "DEVELOPMENT",
      "recent_incumbent_reward_means": [3.896623, 3.911453, 3.795838,
    3.931184, 4.016741, 4.04313]
    },
    "dynamic_mechanism": {
      "factors": [{
          "factor_name": "module_forward_velocity",
          "spatial_scope": "module_local",
          "applicable_active_module_count": 17,
          "active_module_mean": 2.7361,
          "measurement_context": {
            "reference": "voxel_vel", "component": 0,
            "operator": "endpoint_mean", "units": "benchmark_velocity",
            "transform": "identity", "scale": 1.0,
            "verification": "REFERENCE_CHECKED"
          },
          "lowest_module": {"position": [2, 0], "mean": 1.923},
          "highest_module": {"position": [2, 2], "mean": 3.1298},
          "source_summary": {"episode_count": 7,
            "transition_count": 3500, "sample_count": 59500}
      }]
    },
    "trajectory_stage": {
      "spatial_progress": {
        "progress_entity": "robot", "task_axis": "x", "task_direction":
     1,
        "signed_com_displacement_world_units": 4.0431,
        "observation_scope": "REPRESENTATIVE_EPISODE"
      }
    },
    "static_capability": {
      "metrics": {
        "active_voxel_count": 17, "active_body_width": 5,
        "active_body_height": 4, "actuator_count": 4
      }
    }
  },
  "evidence_catalog": [
    {"evidence_id": "E001", "packet_path": "evidence_cards.outcome"},
    {
```

```
      "evidence_id": "E006",
      "packet_path": "evidence_cards.dynamic_mechanism.factors.0"
    }
  ]
}
```


The candidate score, 4.04313, is the mean return on the seven-episode selection panel described in Appendix B. The post-calibration report summarizes the displayed factor over 17 occupied modules and seven recorded episodes, with low/high module entries retaining spatial contrast. Trajectory displacement describes a representative episode, while static entries give body size and actuator count. These scope annotations identify the observations behind each measurement, and evidence IDs such as `E001` and `E006` let the LLM cite them in a later design brief.

Because this report precedes online search, it has no completed intervention or prediction assessment. After a candidate is tested, its report is joined with the saved edit, predicted change, comparison reference, and separate selection and prediction outcomes to form a trial record.

# I Additional Experimental Details

## I.1 Design Traces

Figure 4 follows two Climber and Carrier sequences with complete records of proposals, measurements, and references to earlier trials. Each edit is paired with its measured outcome, showing how that result enters the reasoning for a subsequent proposal.

**Climber.** Run 1, online trials 16–18, predicts higher mean robot $y$-velocity over the last two of five episode bins. Recorded reference/candidate measurements are $0.2397 \rightarrow 0.0102$, $0.2397 \rightarrow 3.1785$, and $3.1785 \rightarrow 3.3376$ in world-frame velocity units. Trials 16 and 17 edit the same selected body; trial 18 edits trial 17's accepted body. Before trial 17, Diagnose cites the failed softer-interface trial and Adapt adds two outer horizontal actuators. The measured increase supports late upward motion; the proposed wall-clamping mechanism remains an interpretation.

**Carrier.** Section 4.3 follows run 5, online trials 1–3. Trial 1 predicts greater robot/object envelope overlap, but the binary indicator remains $0 \rightarrow 0$ despite the score gain. The next two trials shift the prediction target to signed object displacement. Relative to trial 1's accepted body, the measured changes are $6.5196 \rightarrow 1.3754$ and $6.5196 \rightarrow 7.1541$ world units for trials 2 and 3, respectively. These displacement values describe the representative episode. The overlap indicator captures geometric proximity between the robot and object envelopes.

## I.2 Controlled Evidence and Memory Ablations

The ablations examine three information choices in PDA: describing realized behavior, locating it within the body, and retaining the experimental context of earlier edits. Four variants progressively add behavioral measurements and prediction targets, module localization, and richer trial records. Each task uses six search seeds per variant, matching the number of runs per method in the main performance comparison. Table 10 defines the information supplied to each variant; Table 11 gives the common search protocol.

Table 10: Information supplied to Diagnose and Adapt. All conditions retain scores, learning curves, static geometry, and body/edit/score history.

| Variant | Behavior and targets | Module localization | Rich trial records |
|---|---|---|---|
| PDA-S | – | – | – |
| PDA-G | Yes | – | – |
| PDA-L | Yes | Yes | – |
| PDA-R | Yes | Yes | Yes |

**(a) Climber-v0: from softer rails to a wider body**

Run 1 | Trials 16–18

| | Start of trace<br>Before trial 16 | Trial 16<br>Edited from start | Trial 17<br>Edited from start | Trial 18<br>Edited from trial 17 |
|---|---|---|---|---|
| | <br>4-cell-wide body<br>5-cell-wide channel | <br>Soften two<br>upper rail cells | <br>Add two outer<br>horizontal actuators | <br>Add one middle<br>horizontal actuator |
| Panel score | **0.241** | **0.241 → 0.174** | **0.241 → 3.774** | **3.774 → 3.963** |
| Selection | — | **Rejected** | **Selected** | **Selected** |
| Predicted change | — | ↑ Late y-velocity | ↑ Late y-velocity | ↑ Late y-velocity |
| Observed | — | 0.240 → 0.010 | 0.240 → 3.179 | 3.179 → 3.338 |
| Check | — | **Contradicted** | **Supported** | **Supported** |

Before trial 17, memory recalls the failed softening edit; the next proposal tests a wider body.

**(b) Carrier-v0: retaining a gain while revising its explanation**

Run 5 | Trials 1–3

| | Start of trace<br>Before trial 1 | Trial 1<br>Edited from start | Trial 2<br>Edited from trial 1 | Trial 3<br>Edited from trial 1 |
|---|---|---|---|---|
| | <br>Selected initial<br>design | <br>Add a rear wall;<br>cut a deck notch | <br>Add a central leg<br>and a soft foot | <br>Soften two cells<br>in the rigid spine |
| Panel score | **3.247** | **3.247 → 6.564** | **6.564 → 1.386** | **6.564 → 7.192** |
| Selection | — | **Selected** | **Rejected** | **Selected** |
| Predicted change | — | ↑ Envelope overlap | ↑ Signed object displacement | ↑ Signed object displacement |
| Observed | — | 0 → 0 | 6.520 → 1.375 | 6.520 → 7.154 |
| Check | — | **Contradicted** | **Contradicted** | **Supported** |

Before trial 3, memory links trial 1's gain and failed prediction with trial 2's rejection.

Rigid  Soft  Horizontal actuator  Vertical actuator  Edited cell

Figure 4: **Two outcomes of each design intervention.** Each trial names its edit base; arrows show base-to-candidate changes, and black outlines mark edited cells. Panel scores are seven-episode mean returns after 512K training steps. Selection follows performance, while the prediction check assesses the observed behavioral change. Notes summarize how prior trials enter later reasoning. Trials count online candidates; Appendix I.1 details the metrics and records.

**Shared starting points.** For each task and seed, three initial bodies are trained once. All four variants reuse these bodies, the selected controller, learning curves, and calibrated measurements before their information filters become active. This yields $2 \times 6 = 12$ initialization bundles, each containing three trained bodies. Every search is charged its three-body initialization budget, while reused and new computation are logged separately. The four variants then develop their own proposals and histories from this shared starting point.

**Current observations.** All variants receive task/reward semantics, design constraints, static geometry, the current body, panel scores, and learning curves. PDA-S reasons from these inputs and can record an unmeasured hypothesis. PDA-G additionally receives whole-body trajectories, pooled probe statistics, and their definitions as targets for behavioral predictions. PDA-L and PDA-R retain

Table 11: Shared protocol for the information ablations.

| Setting | Protocol |
|---|---|
| Tasks and seeds | UpStepper-v0 and Pusher-v0; six seeds per task and variant ($2 \times 6 \times 4 = 48$ searches). |
| Initialization | Three trained initial bodies; the same selected body, controller, and frozen measurement interface within each task/seed. |
| Online search | 20 candidate bodies, each trained from scratch for 512K steps; no controller-only continuation. |
| Selection evaluation | Frozen selected checkpoint; mean return over the same seven-episode panel configuration in all four conditions. |
| Proposal base | Current design in all four variants. |
| History retrieval | Latest three completed online trials, including failures; all conditions retain a chronological body/edit/score summary. |
| Total charged training | 23 body evaluations, or 11,776,000 steps per complete search, including shared initialization. |

module identities and local contrasts, connecting measurements to editable body components. Static voxel coordinates and legal growth sites remain available throughout.

**Historical information.** PDA-S, PDA-G, and PDA-L receive past bodies, exact edits, panel scores, and comparison context. PDA-R adds hypotheses, pre-trial predictions, measured changes, assessments, and revisable interpretations. This record connects what an edit was intended to change with what happened and how the outcome was interpreted. The PDA-L to PDA-R comparison therefore evaluates the contribution of this experimental history. All four variants use the current design as their proposal base and retrieve the latest three completed trials. The main PDA searches instead use context-ranked retrieval and allow bounded follow-up bases (Appendices F and E). PDA-R thus supplies the richest information within the common ablation protocol.

**Controlled comparisons and reporting.** Each information filter applies consistently to Diagnose, Adapt, and their repair and review calls. Structural validity, novelty checks, checkpoint selection, and score-based acceptance are shared. For each task and variant, Figure 3 reports the mean and SD of search performance (the best panel score found so far) across six seeds, from evaluation 3 to 23. Each plotted curve therefore summarizes six searches with different seeds; variants with the same task and seed share initialization. Initialization and rejected trained candidates both count toward this budget. The successive contrasts connect search progress to the information available for proposing and interpreting the next edit.

## J Understanding the Thrower Gap

Thrower shows the clearest gap to the longer-run baseline endpoints in Figure 2. For the diagnostic analysis here, we examine 152 completed online trials from four logged runs using saved proposals, edits, panel scores, predictions, and diagnoses. These counts describe the diagnostic records analyzed below, separately from the six-search performance aggregation in Figure 2. The analysis uses each candidate's 512K-step trial record.

### J.1 Search Behavior and Measurement Coverage

The task rewards forward payload displacement, with a robot-motion term and a simulator-instability penalty. Obstacles make the trajectory important: horizontal motion can produce little range if the payload encounters a barrier. Upward clearance may help, but the task does not require the robot itself to jump.

Only 14 of 152 candidates replace the current design (Table 12). The functional-consistency assessments are supported in 19 trials, contradicted in 97, and unclear in 36. Of the unclear cases, six lack a verified metric binding, 13 show no change beyond tolerance, and 17 concern auxiliary observations rather than the proposed function. Ten of those auxiliary changes match their predicted

direction and seven oppose it. These categories distinguish missing evidence from an observed but insufficiently informative change.

Table 12: Completed Thrower trials. Prediction columns give functional-consistency assessments. A rigid base means all occupied cells in the lowest occupied row are rigid.

| Run | Trials | Selected | Supported | Contradicted | Unclear | Rigid base |
|---|---|---|---|---|---|---|
| 5 | 33 | 3 | 3 | 18 | 12 | 3/33 |
| 7 | 47 | 3 | 6 | 26 | 15 | 44/47 |
| 10 | 30 | 5 | 7 | 21 | 2 | 29/30 |
| 12 | 42 | 3 | 3 | 32 | 7 | 0/42 |
| Total | 152 | 14 | 19 | 97 | 36 | 76/152 |

The search concentrates both its edits and its measurements. Of the 152 proposals, 118 change only one or two cells. Signed object displacement is the prediction target in 77 trials, and mean forward object velocity after peak speed in another 49. Thus, 82.9% of trials use one of two progress summaries. These choices can support local refinement while leaving alternative interaction strategies poorly distinguished.

Rigid bases occur frequently in runs 7 and 10, but never in run 12, which still selects only three candidates. Base rigidity alone therefore does not explain the gap. Its effect would depend on the particular body, actuation pattern, and learned policy.

### J.2 A Supported Prediction with a Lower Score

Run 10 provides a concrete example. Trial 14 replaces a rigid cell with a horizontal actuator, increasing both the panel score and the predicted post-peak velocity. Trial 16 adds a rigid cell to that accepted body: the same velocity statistic rises further, yet the score falls and the candidate is rejected (Table 13).

Table 13: Prediction support and performance can diverge. Coordinates are zero-based; velocity is measured in raw world-frame units. Trial 16 edits trial 14's accepted body.

| Trial | Exact edit | Score | Post-peak $v_x$ | Decisions |
|---|---|---|---|---|
| 14 | $(1, 1)$: rigid $\rightarrow$ horizontal actuator | $0.860 \rightarrow 1.699$ | $0.241 \rightarrow 0.619$ | Select; supported |
| 16 | $(2, 0)$: empty $\rightarrow$ rigid | $1.699 \rightarrow 0.853$ | $0.619 \rightarrow 3.591$ | Reject; supported |

The statistic averages forward object velocity from peak object speed to episode end. Because the peak occurs at a different time for each body, a larger average can describe a shorter travel interval. In trial 16, the peak occurs 0.667 of an episode later in normalized time than in the reference, while signed displacement decreases by 0.786 world units. The saved measurements therefore support increased velocity over that window, not increased range or a better launch.

Before trial 16, Diagnose already identifies decaying horizontal progress and a downstream obstacle as possible limitations. It nevertheless chooses sustained horizontal drive as the next capability to test. The difficulty lies beyond knowing that an obstacle exists: the available summaries leave several physical explanations compatible with the same observation. The panel score retains the better-performing pair, while the rejected trial contributes another observation to the search history.

### J.3 Implications for Evidence and Memory

Table 14 cross-tabulates the same 152 trials. Fourteen of the 19 supported functional predictions belong to candidates that are not selected. Conversely, nine of the 14 selected bodies have contradicted or unclear predictions. These outcomes motivate separating body selection from prediction assessment: neither label determines the other. Progress measurements can test a declared change without identifying the load path, contact timing, or launch mechanism that produced it.

Table 14: Design selection and functional-consistency assessments across 152 Thrower trials. The row labels include functional relevance as well as metric-direction agreement.

| Functional consistency | Candidate selected | Not selected | Total |
|---|---|---|---|
| Supported | 5 | 14 | 19 |
| Contradicted | 3 | 94 | 97 |
| Unclear | 6 | 30 | 36 |
| Total | 14 | 138 | 152 |

In particular, the ten direction-matching auxiliary observations remain in the unclear row because their functional relevance is unresolved. Keeping this distinction in the record lets later diagnoses separate a measured change from evidence for the proposed capability.

Memory preserves both outcomes for later proposals. In Thrower, however, the concentration of edits and prediction targets means that many records revisit similar interventions and progress summaries. These records provide repeated observations of the tested interventions, while alternative interaction strategies remain less explored. The Carrier trace in Section 4.3 shows a different use of history: the recorded reasoning revisits the explanation for an accepted design, then uses a rejected edit to distinguish the next structural change (Figure 4).

Each trial score describes the body together with the controller learned for it. Equal PPO budgets provide the same training opportunity, and the resulting behavior depends on both the proposed structure and the outcome of controller learning. The Thrower records therefore locate the observed search gap in this joint process: repeated local proposals, similar progress measurements, and one controller-training run for each candidate.

# K UNIMAL Settings and Design Traces

## K.1 Platform Adaptation

UNIMAL tests the same experimental loop on articulated bodies. DERL supplies the design primitives and reference single-body learner; MetaMorph supplies limb-level observations (Gupta et al., 2021; 2022). Each task starts a new search with fresh controllers and memory. Table 15 summarizes the adapter changes.

Table 15: The shared PDA loop with platform-specific representations.

| Component | Evolution Gym | UNIMAL |
|---|---|---|
| Body | Material-labeled voxel grid | Symmetric tree of capsule limbs and hinges |
| Edit | Explicit voxel changes | Typed operation on a symmetry group |
| Probe | Module transition function | Expressions over limb/joint signals |
| Controller | Fresh 64–64 MLP; 512K steps | Fresh 64–64 MLP; about 5M steps |
| Prediction reference | Current design | Actual proposal base |
| Shared procedure | Fixed measurements; evidence-based brief; pre-trial prediction; return-based selection; within-run experimental memory. | |

## K.2 Tasks and Design Space

Both tasks use a fixed $10^\circ$ incline rising along world $+x$, with $+z$ upward. Each action advances four 0.005-s physics steps, and episodes last up to 1000 actions. Reset perturbations are uniform in $[-0.005, 0.005]$ for the configured robot position/velocity entries. Table 16 gives the principal task settings.

Table 16: UNIMAL task settings. The action cost is $c(a) = 0.001 \sum_i a_i^2$, and $h_0$ is initial ground-relative torso height.

| Setting | Incline | Push Box Incline |
|---|---|---|
| Terrain size | $150 \times 40$ m | $80 \times 40$ m |
| Objective | Move uphill | Move a box toward an uphill goal |
| Reward | $v_x / \cos(10^\circ) - c(a)$ | $r_{\text{reach}} + r_{\text{push}} - c(a)$ |
| Fall threshold | Ground-relative torso height $\leq 0.2 h_0$ | |
| Lateral termination | $\lvert y \rvert \geq 19$ m | None |
| Episode limit | 1000 actions | 1000 actions |

Push Box Incline places a 2-kg box of side length 0.40 m at $(-35, 0)$ and its goal at $(35, 0)$, with the robot behind the box. The reach reward is 100 times the decrease in minimum agent–object site distance, plus a one-time bonus of 10 upon reaching 0.5 m proximity. Afterward, the push reward is 100 times the decrease in planar box–goal distance, plus 10 per step within 0.5 m of the goal. Goal proximity does not end the episode. Both tasks have zero standing reward, and clearance is measured relative to terrain rather than world height.

Bodies comprise a spherical torso and 2–10 capsule limbs in a tree. Initialization proposes 4-, 6-, and 8-limb designs. Online edits act on complete bilateral symmetry groups and are compiled into simulator bodies before training. Table 17 lists the permitted values.

Table 17: UNIMAL design space. Physical parameters are selected from discrete grids.

| Field | Values |
|---|---|
| Limb count / radius | 2–10 / 0.05 m |
| Limb length | $\{0.2, 0.3, 0.4\}$ m |
| Torso radius | 0.10 m |
| Limb and torso density | $\{500, 600, 700, 800, 900, 1000\}$ kg/m$^3$ |
| Motor gear | $\{150, 200, 250, 300\}$ |
| Joint axes | $x$, $y$, or both $xy$ |
| Joint limits (degrees) | $[-30, 0]$, $[0, 30]$, $[-30, 30]$, $[-45, 45]$, $[-45, 0]$, $[0, 45]$, $[-60, 0]$, $[0, 60]$, $[-60, 60]$, $[-90, 0]$, $[0, 90]$, $[-60, 30]$, $[-30, 60]$ |
| Operations | Grow/delete limbs; change limb parameters, joint range, degrees of freedom, gear, or density |

Each proposal applies one typed operation; growth or deletion may affect several symmetric limbs. Joint-range changes preserve attachment geometry and neutral pose. Validation checks legality and realized-body novelty. This gives a mixed structural and parametric search space with a coarser edit unit than Evolution Gym's cell edits.

### K.3 Observations and Controller Learning

The MLP receives flattened, masked limb proprioception, with object and goal observations added for Push Box Incline. Module fields include position, velocity, orientation, joint state, geometry, mass, and motor parameters. The adapter pads to 12 limb slots and distinguishes missing entries with masks. Box joints are excluded from robot actions.

Probes use raw signals before controller normalization. Restricted expressions combine arithmetic, absolute value, square root, minimum/maximum, and clipping; static expressions describe the compiled body. Temporal summaries retain event coverage and terrain-relative coordinates. Prediction checks use the proposal base as reference, while return comparisons use the current design.

The controller follows DERL's two-layer MLP and principal PPO settings (Table 18). A nominal 5M-step budget fits 1220 complete batches of 4096 transitions, giving $B = 4{,}997{,}120$. Development evaluations occur every 122 updates and select the checkpoint from this single training run. Its controller and observation normalizer are then frozen for a separate seven-episode selection panel.

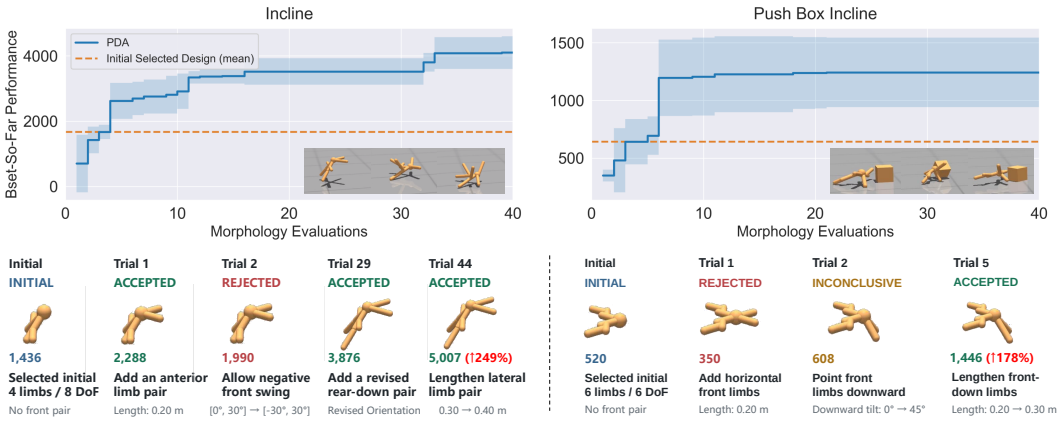


Figure 5: **PDA beyond voxel designs.** Top: mean search performance (best selection-panel score) and SD across six independent searches; dashed lines mark mean initial performance. Bottom: selected edit sequences, including an Incline trial beyond the upper panels' 40-evaluation horizon. Case gains are relative to their own initial designs.

The panel mean supplies the candidate's score, and its trajectories provide behavioral evidence. The resulting scores guide selection and define search performance through Eq. 4.

Table 18: UNIMAL single-body controller settings.

| Parameter | Value |
|---|---|
| Actor / critic | Separate 64–64 MLPs, Tanh |
| Policy | Gaussian; learned log standard deviation |
| Weight initialization | Orthogonal, gain 1; zero biases |
| Initial log std. / bounds | $0$ / $[-5, 2]$ |
| Optimizer / learning rate / Adam $\epsilon$ | Adam / $3 \times 10^{-4}$ / $10^{-5}$ |
| Learning-rate schedule | Linear decay |
| Discount $\gamma$ / GAE $\lambda$ | $0.99$ / $0.95$ |
| PPO clip / epochs / minibatch size | $0.2$ / $4$ / $512$ |
| Value / entropy coefficient | $0.5$ / $0.01$ |
| Value clipping / gradient norm | Enabled / $0.5$ |
| Observation / reward normalization | Enabled; clipped to $[-10, 10]$ |
| Parallel environments / rollout length | $32$ / $128$ |
| Training updates / steps | $1220$ / $4{,}997{,}120$ |
| Development / selection episodes | $5$ / $7$ |

Numerically invalid transitions are excluded from PPO targets and normalization, with their simulator cost recorded. Invalid evaluation panels are ineligible for selection. A failed candidate retains the current design.

### K.4 Reporting and Design Traces

Figure 5 reports search performance through 40 trained morphologies including initialization. At each evaluation count, the upper panels show the mean and standard deviation of the best panel score across six independent searches. The dashed reference averages the scores of their selected initial pairs. Per-run records identify the model, development-checkpoint rule, seeds, and consumed budget.

The lower panels show selected within-run sequences. Their trial indices count online proposals, and intermediate designs are omitted. In particular, Incline trial 44 lies beyond the upper plot's 40-evaluation horizon. Each percentage compares the panel scores of that case's initial and last displayed body–controller pairs.

**Incline.** The initial four-limb, eight-degree-of-freedom body scores 1,436. Adding a 0.20-m anterior limb pair raises the score to 2,288 at trial 1. Expanding the front-swing range from $[0, 30]^\circ$ to $[-30, 30]^\circ$ then lowers the score to 1,990 and is rejected. Trial 29 adds a revised rear-down pair and reaches 3,876; trial 44 lengthens a lateral pair from 0.30 to 0.40 m and reaches 5,007, a 249% gain over initialization.

**Push Box Incline.** The initial six-limb, six-degree-of-freedom body scores 520. Adding horizontal front limbs yields 350 and is rejected. Pointing these limbs downward from $0^\circ$ to $45^\circ$ yields an inconclusive 608. Trial 5 lengthens the front-down limbs from 0.20 to 0.30 m and reaches 1,446, a 178% gain over initialization. This sequence distinguishes changing limb orientation from increasing joint excursion.

All displayed bodies receive fresh controllers at the same per-body budget. Each verdict refers to its recorded comparison parent, which need not be the preceding displayed body. Together, these cases show how the loop applies to articulated edits; the controlled evidence and memory comparisons are provided by the Evolution Gym ablations.