LANGUAGE TO DESIGN (L2D): TOWARDS SELF EVOLVING ROBOTS

Anonymous authors
Paper under double-blind review

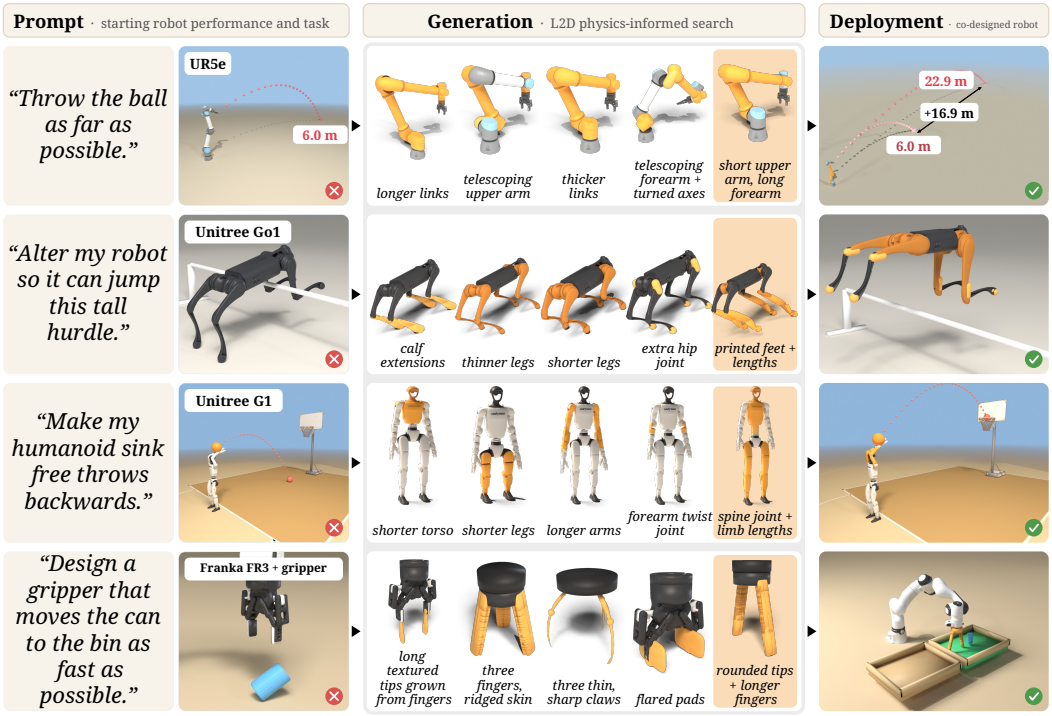


Figure 1: **Prompt to design and control** A prompt with the initial robot is used to measure baseline behavior(left). Using L2D, the robot is optimized and altered with physics guided language reflection. The designs are updated, verified, and ranked with final co-designed robot selected (right).

ABSTRACT

Designing a robot requires expertise in both complex mechanics and control. This process typically relies on slow cycles of estimation, prototyping, and testing to find where design and behavior align. This raises a fundamental question: can robot design be made as simple as describing what we want a robot to do? We present Language to Design (L2D), a co-design framework that turns a prompt and an existing robot into an optimized body, a motion that performs the task, and manufacturable parts, without tuning or training per task or per robot. The framework supports 1) language proposals grounded in measured physics, where language reflection decides what to change and physics measured in simulation decides how much, 2) a continuous design space, spanning link kinematics, motors, added joints, and continuous morphology designs for contact points such as feet and fingers, and 3) real-world fabrication, where prompt with stress-based topology optimization turns added parts into light, manufacturable components can be deployed zero shot on hardware. We demonstrate L2D by co-designing quadrupeds, arms, grippers, dexterous hands, and humanoids across eleven tasks on nine different robots, and deploy a printed design zero shot on a real quadruped for a climbing task. Task videos are available on our anonymous website.

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.

1 INTRODUCTION

Designing a robot takes expertise in complex mechanics, hardware design, dynamics and control (Ha et al., 2018a;b). Even when a desired behavior is easy to describe, finding an aligned body and a controller that achieve it remains a difficult engineering process: users must reason about low-level hardware choices before they can tell whether a design suits the behavior they have in mind (Megaro et al., 2015; Schulz et al., 2017). To this end, we ask: **can robot design be made as simple as describing what we want a robot to do?**

We study prompt-conditioned robot co-design: a user gives a description of the desired behavior and design constraints, the task scene and the initial robot, and receives an improved body and controller, deployable zero shot, without setting parameters or training a controller. One system should **generalize** across robot families (quadrupeds, arms, grippers, dexterous hands and humanoids) and tasks, from dynamic locomotion to in-hand dexterity. This is hard for three reasons: design and control spaces are large and differ between families (Yuan et al., 2022; Bhatia et al., 2021; Zhao et al., 2020; Fay et al., 2026); each task needs its own objective, which the search it guides can exploit (Pan et al., 2022; Gao et al., 2023; Lehman et al., 2020); and every candidate body needs its own motion verified in simulation or the real world before it can be judged, so each evaluation is expensive (Gupta et al., 2021; Yuan et al., 2022).

We introduce *Language to Design (L2D)*, whose central idea is to let a language model propose *what* to change from failure reflection and let physics measured in simulation decide *how much*. From the sentence, the language model writes a dense reward for a training-free planner, constraints on what may change and a first map of where to search. Designs are ranked only by a task judge that the user signs off, fixed before the run and hidden from the planner, and any success outranks any failure, so designs improve on the task itself rather than on a proxy the planner can exploit.

L2D rests on four contributions. First, a **universal, continuous design space** converted directly from any robot's simulation file (MJCF) (Todorov et al., 2012), beyond grammars or primitives: link lengths, joint axes, masses, gears and gains become knobs around the starting robot, contact parts are regenerated as signed distance functions (SDFs), joints can be spliced into any link, and every body plan maps onto one fixed-length vector, from a 240-knob quadruped to an 825-knob humanoid. Second, a **language model grounded in measured physics**, which turns the prompt's constraints (for example, "only add lightweight 3D-printed parts to the lower legs") into a fence on the design space and proposes edits by reasoning about why the current best design fails, which physical quantities limit it and how its earlier predictions fared; because designs share one set of physical quantities, finished tasks give related tasks a better starting point as the design library grows. Third, **physics-grounded design edits**: a small model fitted on the run's own results predicts the score from 33 physical quantities, and a finite-difference Jacobian turns its direction into a least-squares knob edit, giving directed updates from a black-box simulator without differentiable physics (Hu et al., 2020; Xu et al., 2021) or a learned dynamics model. Language leads while data is scarce, physics once the model is accurate, and cheap screens and warm-started planners (Pinneri et al., 2020) focus the motion search. Fourth, **sim-to-real post-processing**: stress-based topology optimization under the design's simulated loads turns added material into light, printable parts that users can adjust.

We evaluate L2D on eleven tasks and nine robots: five main tasks across five robot families (a quadruped hurdle, an arm toss, bin picking with a generated gripper, dexterous pen spinning and a humanoid free throw), three tasks it was not developed on (curling, a rocky slope and hammering a nail), two transfer tasks, and climbing with a real quadruped. The search reaches complex designs, from an added knee joint to a generated foot or a three-fingered gripper. L2D takes a step toward making robot design accessible to non-experts: **describe the task, and let language and physics determine how the robot should be built.** Results available on our website: `https://anonymous.4open.science/w/language-to-design/`.

2 METHOD

Overview. The user provides the robot's simulation file (MJCF), a prompt with the task and any design constraints, and the task scene, and signs off the task judge; L2D returns an optimized design, its motion and, when material is added, printable parts (Fig. 1). A language model (GPT-6 Astra (OpenAI, 2026)) turns the prompt into a design space problem (§2.1), the robot file becomes

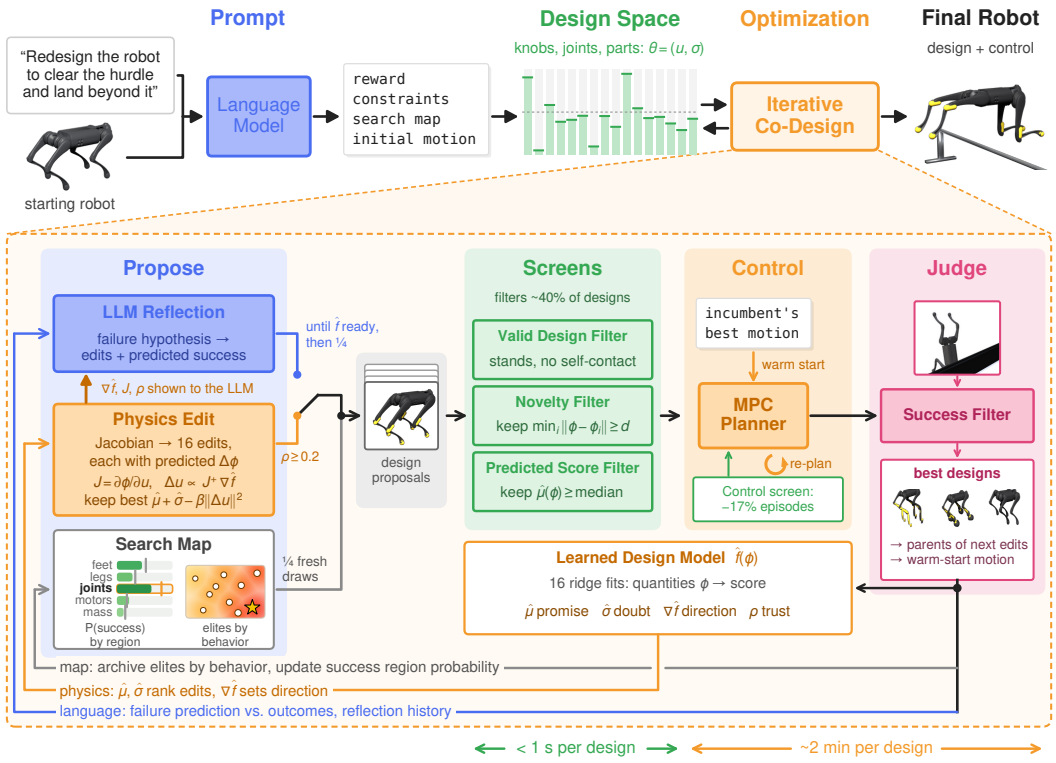


Figure 2: **L2D overview.** *Top:* a language model turns the prompt and the robot's MJCF into a reward, design constraints, a search map, an initial motion and a continuous design space $\theta = (\mathbf{u}, \sigma)$. *Bottom:* one round of the co-design loop (§2.3).

a continuous design space (§2.2), and a co-design loop searches it (§2.3; Fig. 2), in which measured physics informs every language reflection proposal and turns it into precise knob changes. A run lasts until a design or language budget is spent or the user deems the task achieved (App. A.2).

2.1 From a Sentence to a Design Problem

Constraints, reward, and judge. First, the language model reads the prompt, the robot and the scene, turns what the prompt says about the body into limits on the design space (§2.2), and writes the planner's reward in one request that gives the actuators, the scene measured from the robot, the objective, the horizon, the state the reward may read and a code template (App. A.2.2). It states what success physically requires, then writes one reward that pays for its preconditions, with constraints such as "without dropping it" as penalties. The reward only guides the planner; designs are ranked by a task judge of pass/fail clauses (the completion event plus clauses that close known loopholes, such as going around the hurdle) and a graded reading in physical units. The user signs off every clause before the run; the judge is then hashed, hidden from the planner, only made stricter (§2.3.2).

Search map and starting motion. Design regions follow the robot's part groups (whole robot, legs, joints, motors, mass and, on legged robots, feet), each narrowed until most of its random designs do not fail immediately; the first round samples them evenly to seed the search map (§2.3.1). The starting robot's motion comes from the MPC planner under the reward, from scratch or refining a seed motion written by the language model with MPC or as supplied by the user (§2.3.2).

2.2 Design Space Generalization

Continuous in kinematics, shape and topology. L2D reads the MJCF file as generic elements, so one code path handles every robot: it rebuilds 86 of 103 unseen MuJoCo Menagerie models (Zakka et al., 2022) to within 0.1 mm, and a new robot usually needs no hand-written space (97 robots in

nine families, App. A.7). As Fig. 3 shows, every numeric field becomes a knob $u \in [0, 1]$ ($u = \frac{1}{2}$ is stock), kept physically consistent (a longer link is also heavier); contact parts become SDFs with continuous shape parameters; and 43 structural edits change the body itself, up to generating a whole gripper. Every body plan maps onto one fixed-length $\theta = (\mathbf{u}, \sigma)$, from 240 knobs on a Go1 to 825 on a G1, compiled in MuJoCo and settled without control. The loop uses this freedom: the best bin-pick designs were fully generated grippers, and 6 of the 13 best hurdle designs carry a generated foot, a printed calf extension or an added joint.

Constraints and soft limits. The prompt's body constraints (§2.1), such as which parts may change, fence the space: knobs outside it stay at stock, disallowed edits are never drawn, and ranges widen up to four-fold when successful designs crowd a bound (App. A.3). **Physical quantities.** Because hundreds of knobs would need far more designs than a run can afford, L2D describes each design by what its body *is*: 33 physical quantities $\phi(\theta)$ of mass and inertia, posture, geometry and actuation, read from the settled model in a fraction of a second. Structural and knob changes show up in the same quantities, inapplicable ones are zero, and ϕ is z-scored over the run.

2.3 The Co-Design Loop

Each round, the language model reads how the current best design fails, names the likely cause and proposes designs with predicted judge readings; physics edits add designs along measured directions; and after testing, predictions and outcomes are returned to the model, which revises its reasoning (Fig. 2, bottom; App. A.3).

2.3.1 Propose

LLM reflection. GPT-6 Astra proposes designs from the same evidence the physics search uses: how the current best design fails, which quantities the learned design model says should change, which knobs the measured Jacobian says move them most, how far that model can be trusted, the best designs so far, and its own earlier predictions next to what happened. It answers with a short reflection and designs, each with a numeric prediction of the judge's readings, so over a run it learns how far to trust its own physical intuition.

Physics edits. A physics edit moves a design's physical quantities in the direction $\mathbf{g}$, the gradient of the learned design model (§2.3.2). L2D measures by finite differences, without simulating any motion, the Jacobian $J = \partial\phi/\partial\mathbf{u}$ of how each knob changes each quantity, including side effects such as the mass a longer link adds, and takes the smallest knob change that produces the target, $\Delta\mathbf{u} \propto (J^\top J + \epsilon I)^{-1} J^\top \mathbf{g}$, at several step sizes ranked by predicted score and uncertainty. This gives a directed design update from a black-box simulator, with no differentiable physics.

Search map. Design regions (§2.1) are chosen by Thompson sampling on how often their designs have cleared a goal score, and an archive keeps the best design for each kind of behaviour. Its designs are the parents the other two proposers edit, so the search keeps several distinct good designs alive.

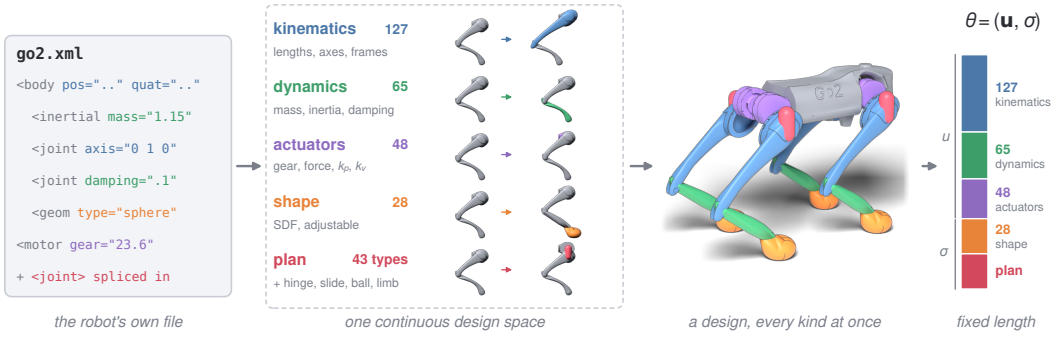


Figure 3: **The design space of a Go2**, read from its own MJCF file (counts per kind).

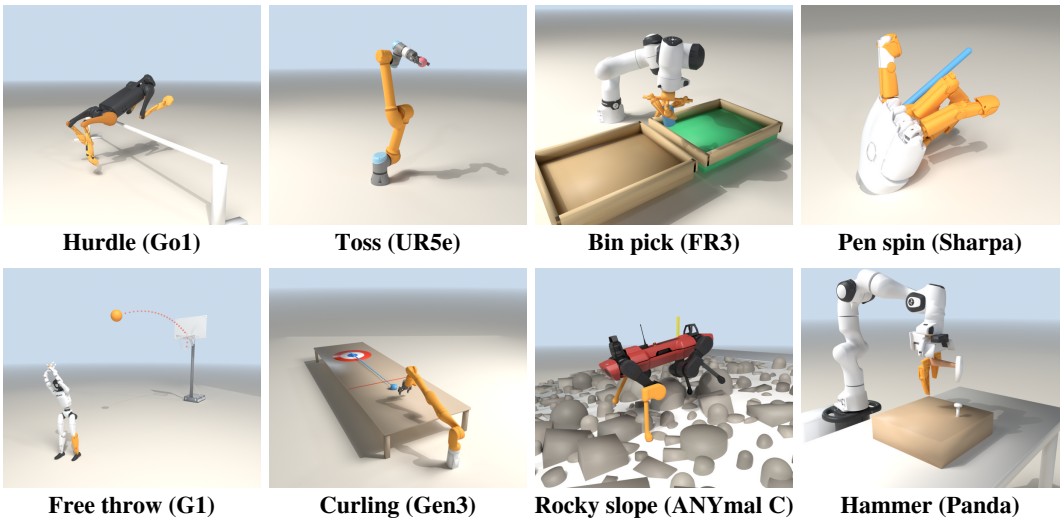


Figure 4: **Task suite.** The five main and three held-out tasks, each with an L2D design (orange: changed parts; held-out poses and the rocky-slope edit illustrative). Additional tasks in appendix.

2.3.2 From Proposal to Score

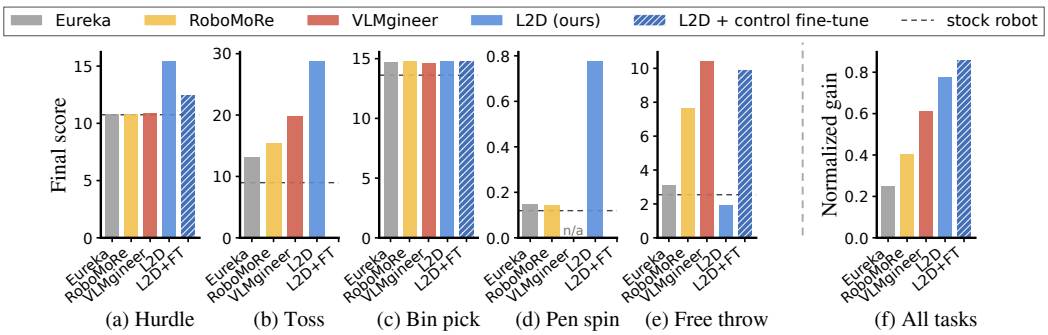
Design proposals. Each round proposes $N = 10$ designs; a quarter are drawn fresh from the search map and the rest are edits. The language model proposes the edits until the learned design model predicts held-out designs (rank correlation ≥ 0.2), after which physics edits propose most of them and the language model returns every fourth round. Language thus supplies ideas while data is scarce, and physics supplies precision once data is available.

Design screens. Finding a motion is the expensive step (100–200 s per design), so three cheap screens run first: a *validity* screen rejects designs that fall over or intersect themselves, a *novelty* screen skips designs too close in physical quantities to one already tested, and, once the learned model is trusted, a *predicted-score* screen keeps the better half of $2N$ candidates. Together they discard about 40% of proposals before any motion is searched at about 1% extra wall-clock time; world models screened no better (App. A.6).

Control. Every remaining design gets its own motion from closed-loop cross-entropy model predictive control on the reward (App. A.2.3); no policy is trained. Each planner starts from the best motion so far, which carries over because editing the body never changes the command interface; this warm start cut total wall-clock time by about 27%. All designs share the planner's random numbers, so score differences are differences between designs. A *control screen* stops searches that fall below the median of earlier designs and skips further starts for designs outside the top 40%, giving the same final designs with 17% fewer simulated episodes.

Judge and score. Designs are ranked by the signed-off judge (§2.1), never by the reward: a successful episode scores 10 plus its graded reading, and a failed one the fraction of the task it completed, so any success outranks every failure. Designs are scored on one starting state, each round's best on four, and the run's three best on six; the highest is the final robot, tested on six held-out starts.

Learned design model. A small linear model is used so after a few dozen designs it can be useful: an ensemble of 16 ridge regressions ($\lambda = 1$), each fitted on a bootstrap resample of the run's designs, that maps the 33 z-scored physical quantities to the judge score and up to three graded readings. Its weights say which quantities go with success on this task (for example, a lower trunk and more torque per kilogram). It is refitted every round from this run alone. Its mean $\hat{\mu}$ ranks candidates, its spread $\hat{\sigma}$ marks where it is unsure, its gradient $\mathbf{g}$ steers the physics edits, and its held-out accuracy decides when physics takes over; every design not filtered is simulated and judged.



(a) Hurdle (b) Toss (c) Bin pick (d) Pen spin (e) Free throw (f) All tasks

Figure 6: **L2D against language-driven co-design at the same language budget.** (a)–(e) Mean final score on six held-out starts (dashed: stock robot). (f) Gain over stock, normalized to the best method per task and averaged (Table 2).

2.4 From Simulation to Hardware

When the final design adds material, L2D replays the winning motion, records the loads on each printed part and runs stress-based topology optimization in printed PLA, keeping the lightest part within its stress and deflection limits (Fig. 5; App. A.10). A user can add simple constraints, such as where the part may attach or which regions must stay clear, without rerunning the search; the part is then printed and deployed zero shot.

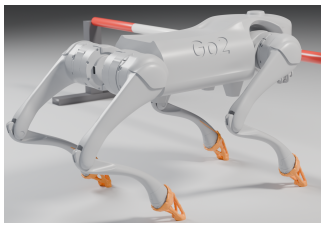


Figure 5: Optimized printed feet (orange) on a Go2.

3 Experiments

We ask four questions. (i) At the same language budget, does L2D find better robots than published language-driven co-design methods and starting robots? (ii) Without language, how does its search compare with black-box design optimizers? (iii) What does each part of the loop contribute? (iv) Do common desired physics behaviors carry over to additional tasks, and do designs transfer between tasks? Study parameters are in App. A.4.

3.1 Setup

Tasks. We test L2D on eleven tasks and nine robots (Fig. 4; App. A.1). All methods share the simulator (MuJoCo 3.11 (Todorov et al., 2012)), each task's design space, controller settings and frozen judge. We report the final robot's score, its gain over the stock robot, and the *pass rate*, the share of simulated designs that succeed (protocol in App. A.4).

Baselines. *Language-driven co-design.* RoboMoRe (Fang et al., 2025) runs a coarse diversity stage and then alternates morphology refinement with reward rewriting. VLMgineer (Gao et al., 2026) shows the model renders of the current best and elite designs and asks for mutations and crossovers. Eureka (Ma et al., 2024) writes and selects rewards on the stock robot and flies the winner on random design edits. All three use the same model as L2D (GPT-6 Astra, low reasoning effort), the same $3.33 budget per seed, and, as in their original form, no design filters. *Black-box design search.* Random search draws designs uniformly from the design space; Bayesian optimization (Snoek et al., 2012) fits a Gaussian process with a Matérn-5/2 kernel and maximizes expected improvement over 512 candidates; CMA-ES (Hansen & Ostermeier, 2001) starts from the stock design. Each runs 1,000 designs per seed with a cold controller and no language model.

4 Results

Comparison with Language-Driven Co-Design. Figure 6 compares L2D with the three language-driven methods at the same language budget. L2D finds the best robot on three of five tasks: on the hurdle its final robots score 15.5 against at most 10.9 for the baselines, whose designs

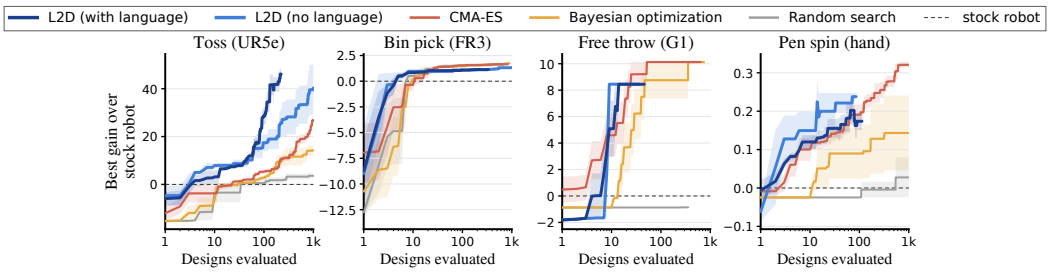


Figure 7: **Design search without language.** Best gain over the stock robot so far (mean over seeds, ± 1 s.e.; L2D curves end early where its validity screen rejects proposals).

barely improve on the stock Go1; on the toss they throw 18.8 m against 9.8 m for the best baseline; and on pen spin they turn the pen furthest, although no method completes the turn. Bin pick saturates at about 14.8 for every method. Averaged over tasks (best method 1), L2D's gain is 0.78 against 0.61 for VLMgineer, 0.40 for RoboMoRe and 0.25 for Eureka. On the free throw, L2D is limited by control: a longer control search on its designs for tasks with complex whole-body control raises their mean from 1.9 to 9.9 (App. A.4).

Comparison with Black-Box Design Search. Without language, we compare L2D's physics search with random search, Bayesian optimization and CMA-ES at 1,000 designs per seed (Fig. 7; Table 5). On the toss, L2D's best design gains 40.3 m over the stock arm against 26.9 m for CMA-ES, 14.2 m for Bayesian optimization and 3.6 m for random search, and matches CMA-ES's final best within 232 designs. On bin pick 93% of its designs succeed against 62 to 70%. CMA-ES is the strongest baseline, gaining most on pen spin and the free throw; averaged over tasks, CMA-ES reaches 0.93, L2D 0.84, Bayesian optimization 0.64 and random search 0.23. Adding language speeds up the toss: after 128 designs L2D with language has gained 35 m against 19 m without, which the language-free search needs about 770 designs to match.

What Designs L2D Finds. Figure 9 shows every design L2D simulated in the language-driven comparison. On every task the search spreads into several distinct families of bodies that differ in how often they succeed, rather than refining one neighbourhood of the stock robot. The most successful families use the structural freedom of the design space, generated feet on the hurdle (90% and 79% of designs clear the board) and generated four-finger grippers on bin pick (all succeed), not rescaled stock parts. On the toss, the family of the best design (a 32.9 m throw) succeeds on 76% of its tries, against 24% for one that changes shoulder mass; on pen spin and the free throw no family succeeds reliably, and the best designs reach 0.35 turns and 73% of a shot (App. A.5).

Real World Deployment: We deploy the 3D-printed additive design for climbing an obstacle too tall for the stock Go2. The prompt allows only added material, because the fixed-length belts driving the Go2's legs are hard to alter on hardware. The design is optimized, printed and deployed with its controller zero shot (videos: `https://anonymous.4open.science/w/language-to-design/`).

Does Language Use the Physics Feedback? We read all 237 language-model answers that followed physics feedback in the language-driven comparison (3,924 designs over 15 runs). Of the 1,431 target quantities the model named, 86% were among those it was shown and 97% of these pointed the same way as the learned model, and 43% of its designs move a knob the Jacobian listed, against 23% by chance, most clearly on the toss (83% against 38%) and pen spin (58% against 27%); in an earlier loop, language beat the language-free loop only when shown a learned design map (App. A.5). With the ablation (§4), this suggests language helps most when grounded in physics.

Ablations. We remove one part of the loop at a time, without language unless stated, on the toss (60 designs per seed) and the hurdle (80; Table 3).

On the toss (Fig. 8), the *control warm start* matters most: without it the best throw halves, from 14.2 to 7.2 m, on every seed.

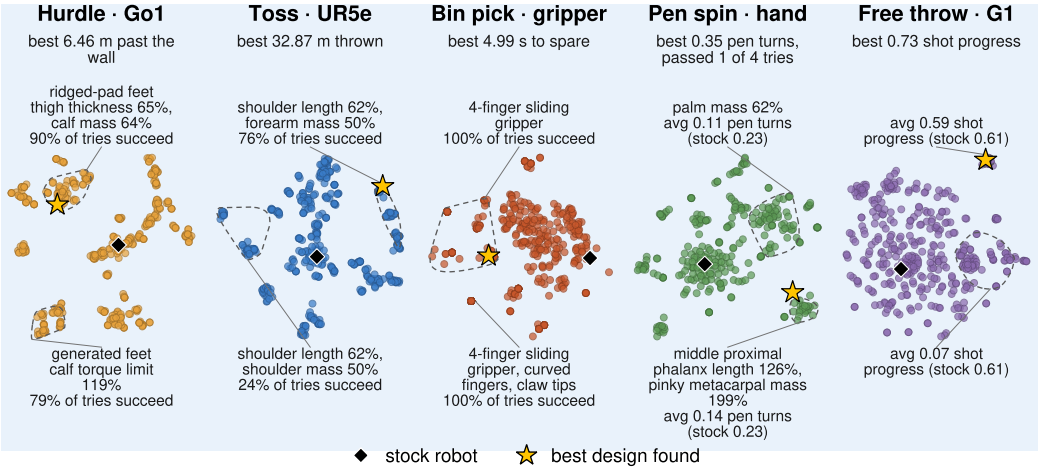


Figure 9: **Designs L2D simulated in the language-driven runs**, t-SNE of physical quantities, knobs and generated parts; outlined clusters are labelled with their shared changes and success rate.

The *physics edits* come next: replacing them with random draws and archive mutations lowers the best throw to 8.5 m. Letting the *language model* propose every edit instead raises the floor but not the ceiling: 83% of its designs pass against 57%, and its mean throw is the highest, but it never proposes the 15 to 20 m designs the physics edits find, so its best throw is 9.8 m, which is why L2D hands over to physics once the learned model is trusted. The *design screen* adds a little (+0.3 m best) and the *novelty rule* is inert on the toss. On the hurdle, where the stock robot already clears, the design screen and the warm start raise the share of designs that clear (0.39 against 0.27 and 0.29; App. A.5).

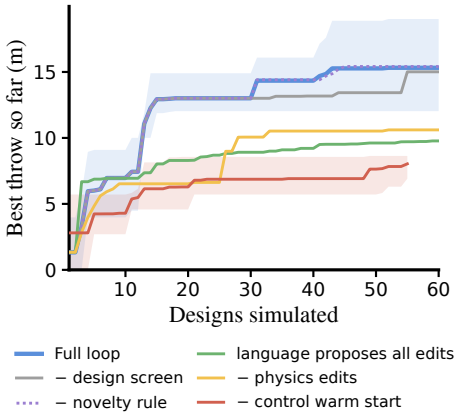
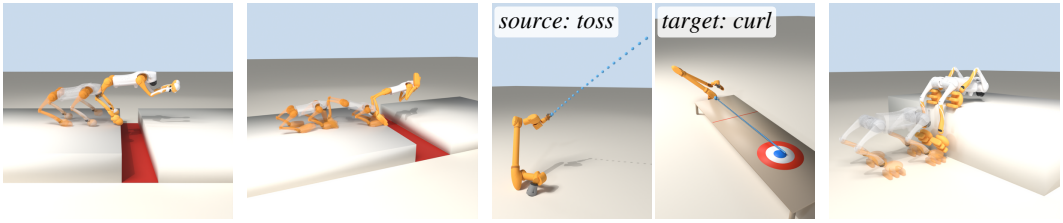


Figure 8: **Ablation on the toss.** Best throw so far with one part of the loop removed (mean of three seeds; bands: range).

Additional Tasks. To test whether the loop carries over, we ran it unchanged, with language, on three tasks it was not developed on (three seeds each, 26 to 139 simulated designs per run, under $1 of language model per run). The stock Panda gripper never drives the nail, while the final designs do on 3 of 18 held-out starts (one per seed). On curling, counting a stone that leaves the ice as a miss, the final designs score 1.57 on average against 0.96 for the stock arm and rest in the house on 4 of 18 held-out starts against 1. On the rocky slope, where the stock robot already reaches the mark, the best design (a longer hind shank) scores 10.30 against 6.82 on its selection starts, but L2D's finals give no reliable gain on held-out starts (11 of 18 against 13).

Transfer Across Tasks. We tested whether designs found on one task help on a related one by flying the source task's two best designs in the first round of a new run (Fig. 10; same robot, three seeds, 80 designs per run; App. A.5). On a Go2, long-jump designs helped a run-up pit leap (different style of jump) on every seed (best design better by 0.45 to 1.29), and the only design that crossed the pit came from a seeded run. Without language, the reverse transfer is stronger: pit-leap designs raised the long jump's final score by 5.3 on every seed, and using the source runs only as a prior for the learned design model and search map raised it by 4.3. Transfer helped where source and target share physics (the two Go2 jumps) and did little where they do not (toss to curling) or the source found nothing (a Go1). Because every design is described by the same physical quantities, finished runs leave designs and results that related tasks can reuse, so as the library grows, more new tasks will start near a good design. We plan to test common physics-behavior task scaling in future work.



Standing long jump (Go2, source) | **Run-up pit leap (Go2, target)** | **Toss → curling (UR5e, source → target)** | **Box climb (Go2, printed feet)**

Figure 10: **Transfer and the real-world task.** Transfer within one robot (Go2 long jump to pit leap; UR5e toss to curling) and the box climb with printed feet (dots: ball and stone paths).

5 Related Work

LLM-driven robot design. Large language models (LLMs) now search over robot designs, mutating designs within an evolutionary loop (Song et al., 2025; Qiu et al., 2026; Gao et al., 2026), jointly rewriting morphology and reward (Fang et al., 2025; Qiu & Cygan, 2026) or turning a prompt directly into a robot (Ringel et al., 2025). However, the model's intuition decides both *what* to change and *by how much*, although foundation models still struggle with physical reasoning (Chow et al., 2025) and optimization by prompting degrades as problems grow (Yang et al., 2024). These methods also work in small or fixed design spaces (Gym creatures (Fang et al., 2025; Qiu & Cygan, 2026), voxel grids (Song et al., 2025), part grammars (Qiu et al., 2026) or one template (Ringel et al., 2025)), and LLM-written rewards (Ma et al., 2024; Xie et al., 2024; Yu et al., 2023) are proxies the search can exploit (Gao et al., 2023; Lehman et al., 2020). L2D grounds language in measured physics and ranks designs by a fixed judge.

Co-design of morphology and control. Prior works grow creatures from blocks, voxels or part grammars (Sims, 1994; Bhatia et al., 2021; Wang et al., 2019; Zhao et al., 2020), co-optimize design and control with reinforcement learning (Luck et al., 2019; Yuan et al., 2022; Lu et al., 2025) or use differentiable simulation (Xu et al., 2021; Hu et al., 2020), each in a benchmark-specific design space and most with a new policy per task. Morphology-conditioned controllers amortize this cost (Strgar & Kriegman, 2025; Ha et al., 2026; Fay et al., 2026) but are tied to the hand-parameterized spaces they were trained on: Transformer Transformer distills reinforcement learning experts at 16 A100-hours each (Ha et al., 2026), and House of Dextra trains a policy per hand family (Fay et al., 2026). L2D is training-free, deriving its space from any robot's simulation file.

Physics-guided design search. Black-box optimizers (Bayesian optimization (Snoek et al., 2012; Eriksson et al., 2019), CMA-ES (Hansen & Ostermeier, 2001), quality diversity (Cully et al., 2015; Vassiliades et al., 2018; Gaier et al., 2017)) see a design only as a parameter vector and a score, and gradients through contact-rich simulation can be unreliable (Suh et al., 2022); L2D learns which physical quantities matter and which parameters move them.

6 Conclusion and Limitations

We presented L2D, which co-designs a robot's body and motion from a prompt by letting a language model reflect on why designs fail and measured physics decide how to change them; it finds better robots than language-driven co-design at the same budget, and a higher ceiling than black-box search where the physics is informative. Two lessons stand out: language alone raises the floor but not the ceiling, and the judge must be fixed and checked for loopholes, since a written reward is a proxy the search can exploit. L2D is limited where success depends on control more than body (pen spin, the free throw), where its physical quantities say little and CMA-ES does better, and by evaluation in simulation. Growing a design library for cross-task transfer and more real-world experiments are the next steps. Additional future work is needed to test further generalization to more tasks and robots.

REPRODUCIBILITY STATEMENT

The design space and how it is built from a robot file are specified in §2.2 and App. A.2; every setting of the co-design loop (proposals, physics edits, screens, planner, judge and learned design model) is in App. A.3, the per-task planner settings in App. A.2.3, the task sentences, design spaces and judges in App. A.1, and the evaluation protocol, budgets, seeds and compute of every study in App. A.4. The language model's prompts and replies, the rewards, the task judges and the per-design results of every run are logged, and we will release the code with them.

AI USE STATEMENT

Language models were part of the method: GPT-6 Astra writes rewards, design constraints and starting motions and proposes designs in L2D and in the language-driven baselines, and proposes designs in the with-language design-search runs (§3). Beyond the method, AI coding and writing assistants helped with plotting code, monitor launched experiments, rendering figures in blender, and draft and edit text. The authors designed the research direction and framework, checked the reported results against the logged runs, and reviewed all AI-assisted work, and take responsibility for the final content of this paper (including text, claims and artifacts produced with the aid of generative AI).

REFERENCES

Jagdeep Singh 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 (NeurIPS)*, 2021.

Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. In *International Conference on Learning Representations (ICLR)*, 2025.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 2015.

David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Jiawei Fang et al. RoboMoRe: LLM-based robot co-design via joint optimization of morphology and reward. *arXiv preprint arXiv:2506.00276*, 2025.

Kehlani Fay, Darin Djapri, Anya Zorin, James Clinton, Ali El Lahib, Hao Su, Michael T. Tolley, Sha Yi, and Xiaolong Wang. House of dextra: Cross-embodied co-design for dexterous hands. In *International Conference on Learning Representations (ICLR)*, 2026.

Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Data-efficient exploration, optimization, and modeling of diverse designs through surrogate assisted illumination. In *Genetic and Evolutionary Computation Conference (GECCO)*, 2017.

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 (ICLR)*, 2026.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*, 2023.

Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature Communications*, 2021.

Huy Ha, Karen Liu, and Shuran Song. Transformer transformer: A unified model for motion-conditioned robot co-design. In *Conference on Robot Learning (CoRL)*, 2026.

Sehoon Ha, Stelian Coros, Alexander Alspach, James M. Bern, Joohyung Kim, and Katsu Yamane. Computational design of robotic devices from high-level motion specifications. *IEEE Transactions on Robotics*, 34(5):1240–1251, 2018a. doi: 10.1109/TRO.2018.2830419.

Sehoon Ha, Stelian Coros, Alexander Alspach, Joohyung Kim, and Katsu Yamane. Computational co-optimization of design parameters and motion trajectories for robotic systems. *The International Journal of Robotics Research*, 37(13-14):1521–1536, 2018b. doi: 10.1177/0278364918771172.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 2001.

Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Fredo Durand. DiffTaichi: Differentiable programming for physical simulation. In *International Conference on Learning Representations (ICLR)*, 2020.

Joel Lehman, Jeff Clune, Dusan Misevic, et al. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial Life*, 2020.

Haofei Lu, Zhe Wu, Junliang Xing, Jianshu Li, Ruoyu Li, Zhe Li, and Yuanchun Shi. BodyGen: Advancing towards efficient embodiment co-design. In *International Conference on Learning Representations (ICLR)*, 2025.

Kevin Sebastian Luck, Heni Ben Amor, and Roberto Calandra. Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

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

Vittorio Megaro, Bernhard Thomaszewski, Maurizio Nitti, Otmar Hilliges, Markus Gross, and Stelian Coros. Interactive design of 3d-printable robotic creatures. *ACM Transactions on Graphics*, 2015.

OpenAI. Gpt-6 astra, 2026. VERIFY: correct citation for the model's zero-shot control ability.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations (ICLR)*, 2022.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning (CoRL)*, 2020.

Kevin Qiu and Marek Cygan. Debate2create: Robot co-design via multi-agent LLM debate. In *International Conference on Machine Learning (ICML)*, 2026.

Kevin Qiu, Władysław Pałucki, Krzysztof Ciebiera, Paweł Fijałkowski, Marek Cygan, and Łukasz Kuciński. RoboMorph: Evolving robot morphology using large language models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Ryan P. Ringel, Zachary S. Charlick, Jiaxun Liu, Boxi Xia, and Boyuan Chen. Text2robot: Evolutionary robot design from text descriptions. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Adriana Schulz, Cynthia Sung, Andrew Spielberg, Wei Zhao, Robin Cheng, Eitan Grinspun, Daniela Rus, and Wojciech Matusik. Interactive robogami: An end-to-end system for design of robots with ground locomotion. *The International Journal of Robotics Research*, 2017.

Karl Sims. Evolving virtual creatures. In *Proceedings of SIGGRAPH*, 1994.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

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 (ICLR)*, 2025.

Luke Strgar and Sam Kriegman. Accelerated co-design of robots through morphological pretraining. *arXiv preprint arXiv:2502.10862*, 2025.

Hyung Ju Terry Suh, Max Simchowitz, Kaiqing Zhang, and Russ Tedrake. Do differentiable simulators give better policy gradients? In *International Conference on Machine Learning (ICML)*, 2022.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation*, 2018.

Tingwu Wang, Yuhao Zhou, Sanja Fidler, and Jimmy Ba. Neural graph evolution: Towards efficient automatic robot design. In *International Conference on Learning Representations (ICLR)*, 2019.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Jie Xu, Tao Chen, Lara Zlokapa, Michael Foshey, Wojciech Matusik, Shinjiro Sueda, and Pulkit Agrawal. An end-to-end differentiable framework for contact-aware robot design. In *Robotics: Science and Systems (RSS)*, 2021.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*, 2024.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning (CoRL)*, 2023.

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 (ICLR)*, 2022.

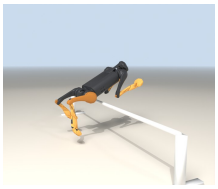
Kevin Zakka, Yuval Tassa, and MuJoCo Menagerie Contributors. MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo. `http://github.com/google-deepmind/mujoco_menagerie`, 2022.

Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. RoboGrammar: Graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics*, 2020.

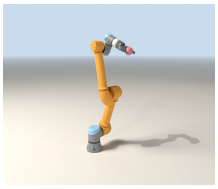
A Appendix

A.1 Task Suite

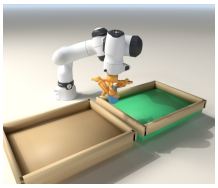
Each task, with a render of an L2D design and its result; Table 1 gives the main tasks' sentences, design spaces and judges.



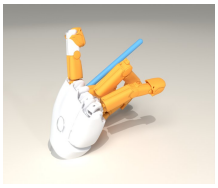
Hurdle (Go1). "Hurdle over the wall in front of you: get over it and past it." The Go1 starts 0.60±0.10 m behind a 0.30 m board and must go over it, not round, and end 5 cm past it (240 knobs, generated feet, printed calf parts, added joints). L2D's finals score 15.5 against at most 10.9 for the language baselines.



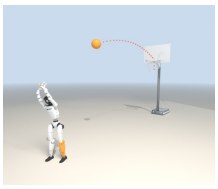
Toss (UR5e). "Throw the ball as far as you can." A UR5e with a Robotiq 2F-85 (119 knobs and joint splices) is graded on the landing distance after a real release. Its finals throw 18.8 m on average, against 9.8 m for the best language baseline.



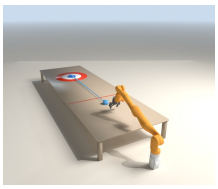
Bin pick (FR3). A Franka FR3 with a generated gripper (40 continuous and 10 discrete parameters) moves a can between two bins and leaves it upright. Every method saturates near 14.8; L2D has the highest pass rate (0.93).



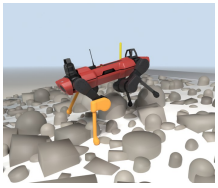
Pen spin (Sharpa hand). "Spin it one full turn without dropping it, ending steady" (397 knobs, fingertip joints and shapes). No method completes the turn; L2D's finals turn the pen furthest (0.78 against at most 0.15).



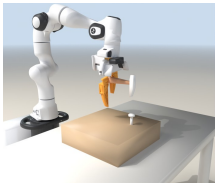
Free throw (G1). "Shoot it into the hoop without stepping over the line" (420 knobs, arm and spine splices, segment lengths). L2D's designs are control-limited: a longer control search raises their mean from 1.9 to 9.9, just below VLMgineer's 10.4.



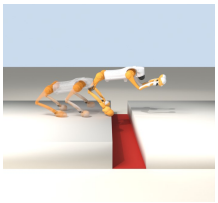
Curling (Gen3). A Kinova Gen3 slides a stone to rest in the house (140 knobs), held out while L2D was developed. With stones that leave the ice counted as misses, finals score 1.57 against 0.96 and rest in the house on 4 of 18 held-out starts against 1.



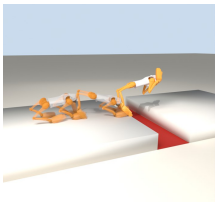
Rocky slope (ANYmal C). An ANYmal C sprints up a rocky slope (240 knobs). The stock robot already reaches the mark; the best design (a longer hind shank) scores 10.30 against 6.82, but finals give no reliable gain on held-out starts (11 of 18 against 13).



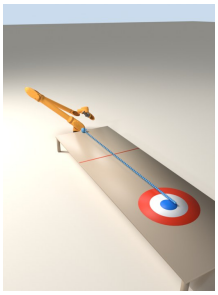
Hammer (Panda). A Panda gripper drives a nail (52 knobs, fingers and mount). The stock gripper never does; the final designs do on 3 of 18 held-out starts.



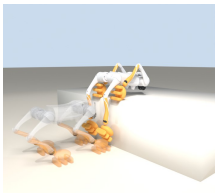
Standing long jump (Go2). A Go2 leaps a 0.30 m pit from standing, with no run-up. The source design for transfer passes on 5 of 6 starts (score 9.45).



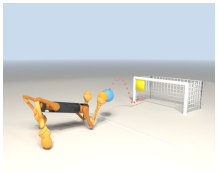
Run-up pit leap (Go2). A Go2 runs up and leaps a 0.58 m pit. Seeding with long-jump designs raised the best design on every seed, and the only design that crossed (2 of 6 starts) came from a seeded run.



Toss to curling (UR5e). The UR5e's curling runs seeded with toss designs, counting a stone that leaves the ice as a miss, end ahead on two of three seeds and tie on the third: weak evidence of transfer.



Box climb (Go2). A Go2 with printed tread feet over its stock calves climbs onto a 0.45 m box and stands on all four feet. The design climbs from 5 of 6 starts, scoring 10.12 against 0.05 for the stock robot; it is the design prepared for hardware.



Soccer (Aliengo). "Shoot this soccer ball into the top left hand corner of the net without crossing the line." A 50-design run without language or physics edits finds enlarged printed feet and retuned legs that score 0.86 against 0.20 and reach the net from 5 of 6 starts (none for stock), though never the top left corner.

Table 1: **Task suite.** Knobs: total → in scope. Episodes are in planner knots (× knot duration).

Task	Robot (family)	Sentence	Knobs	Structural channel	Judge (graded)
Hurdle	Go1 (quadruped)	"hurdle over the wall in front of you: get over it and past it"	240 → 240	leg plans (2nd knee, pogo, twist, hip2, splay); SDF feet (26 params / presets); printed calf extensions and sleeves	root ends $\geq$ 5 cm past a 0.30 m board, went over not round, started behind (distance past, m)
Toss	UR5e + 2F-85 (arm)	"throw the ball as far as you can"	299 → 119	arm joint splices (7 plans)	first touchdown after a real release, in lane, no self-contact (forward m)
Bin pick	FR3 + generated hand (gripper)	"pick the can up out of the bin … put it down inside the green bin"	40 cont. + 10 cat.	generated gripper: 2–4 fingers, 1–3 phalanges, hinge/slide, SDF fingers, skins (1.7×10^6 discrete combos)	in target bin, released, still, upright $\leq$ 20°, hand clear (time to spare, s)
Pen spin	Sharpa hand	"spin it one full turn without dropping it, ending steady"	423 → 397	fingertip joints (knuckle/yaw/slide/twist); SDF fingertips (14 params / 37 presets)	$\geq$ 0.98 net turns while held, never dropped, still at end (turns)
Free throw	G1 + hands (humanoid)	"shoot it into the hoop without stepping over the line"	825 → 420	arm/spine splices; segment lengths (mesh rebuild); arm motors	ball descends through the rim plane after a release, root advanced $\leq$ 1 m, no fall (progress)

A.2 Implementation Details

This section gives the precise form of each component in §2.

A.2.1 One Continuous Design Space for Every Family

Parsing. We parse an MJCF losslessly into a typed list of elements (bodies, joints, geoms, meshes, shapes, equalities, tendons and actuators, each with a pointer to its parent). The parse is deterministic, with nothing learned. It round-trips exactly on all 31 starting robots and on 86 of 103 unseen MuJoCo Menagerie models.

Kinematics and dynamics as continuous knobs. A *knob* is one numeric field of one element: link offset, tilt, joint axis, joint range, mass, inertia, damping, armature, gear, force limit, servo gains, link cross-section or link length. Each knob $u \in [0, 1]$ is centred on the starting robot. Multiplicative fields use

$$x(u) = x_0 \, \lambda^{2u-1}, \qquad \lambda > 1, \tag{1}$$

so $u = \frac{1}{2}$ is the starting robot exactly and equal steps are equal ratios; offsets and rotations are additive about x_0. Lengthening a link scales its mass by s and its inertia by s^3 and moves its children. The design space is thus *generated* from the kinematic tree: 240 knobs on a Go1, 299 on a UR5e, 423 on a Sharpa hand and 825 on a G1 humanoid.

Morphology as continuous solids. Each contact part is regenerated as an SDF with continuous parameters. Feet and fingertips are a smooth union of superquadrics over 28 bounded parameters (reach, taper, sole from plate to cup, lobes, ridges, studs, curl); gripper fingers are tubes swept along a curved centreline with a cross-section, a structural form and a contact skin. Each solid is meshed by marching cubes, collided as convex pieces, and given its mass from its volume. It is stored as one `shape` element of 27 fields.

Topology with continuous parameters. One operator, `insert_dof`, adds a joint to a link: a second knee, a telescoping shank, a wrist or spine joint, or a fingertip knuckle. The added joint's placement, range and gear are themselves knobs. On meshed machines, link lengths are rebuilt continuously by the transform $S = I + (s - 1)\mathbf{a}\mathbf{a}^\top$ about the joint axis $\mathbf{a}$, with consistent inertia ($m' = sm$, $\Sigma' = s S \Sigma S^\top$). Every body plan is lifted back onto the starting robot's knob vector by element name, so topology, kinematics and morphology move in one fixed-length vector per task.

A.2.2 From Prompt to Reward, Design Constraints and a Search Map

Reward. A language model writes a dense reward r_ℓ once, as Python over a fixed state interface (trunk, limb tips, scene bodies, contacts, phase). It sees the sentence, the machine, and the scene as measured from the robot. The sentence's task constraints ("without stepping over the line", "without dropping it") become reward terms. A reward is kept only if a search on the starting robot scores higher under it than standing still, and it is then frozen. Designs are ranked separately, by a judge

J that reads world facts off each episode: a pass scores $10 + g$ in the judge's units g, a failure its progress $\tilde{g} < 1$.

Design constraints. The same model reads the sentence's constraints on the body: which parts may change, whether material may only be added (e.g. lightweight printed parts over the stock legs), and which structural edits are allowed. It turns them into a fence on the design space. Knob groups outside the fence are held at the stock value and disallowed body plans are never drawn. Under an additive fence every design is the stock part plus a printed add-on. Without constraints, the editable part is read from the kinematic tree: arm tasks edit links above the end-effector root, gripper tasks the links below it, and legged robots and hands the whole machine.

Initial search map. The fenced space is partitioned into *regions* along element groups. On a quadruped, for example, these are feet, leg shape, kinematics, leg topology, motors and mass. Before search, each region's sampling width is calibrated, shrinking it until at least 75% of its draws are valid bodies. Round 0 spreads its draws evenly across regions, and its designs seed the archive (App. A.3, *Search map*).

A.2.3 The Co-Design Loop

Each round proposes bodies, screens them, flies them, judges them and reflects on the outcome (Alg. 1); the settings of each step are in App. A.3. The starting robot's planner is seeded with one zero-shot motion written by Astra (OpenAI, 2026).

Controller settings per task. Unless stated otherwise the planner uses 96 samples, four iterations, the best quarter as elites, an initial standard deviation of 1.0 of the command range (floor 0.05), mean smoothing 0.3, and knots held between control steps. Control steps last 0.04 s on quadrupeds and 0.048 s on arms, hands and the humanoid. In closed-loop tasks the plan is shifted after each re-plan and its new tail is reset to the prior. "Delta" searches smooth offsets added to the warm-start motion rather than the motion itself, and "elitist" carries the best sequence found so far into every iteration.

A.3 Co-Design Loop Details

This section gives the low-level settings of the loop summarized in §2.3.

Soft knob limits. When at least half of the 12 most recent successful designs sit within 12% of a bound of the same knob group, that group's range widens by 25%, up to four-fold.

LLM reflection. GPT-6 Astra (low effort, capped response length, about \$0.15–0.30 per call) reads the task, the constraints and the design space (every knob in $[0, 1]$, 0.5 stock), how the current best design fails, three blocks of physics feedback, the best designs so far, and its own earlier predictions next to the outcomes. The feedback blocks are the quantities the learned model says must change (signed, in z units), for each the five knobs the measured Jacobian says move it most, and the model's held-out rank correlation for the score and each graded reading. It answers with a reflection and designs, each naming a parent, the knobs, generated parts or added joints to change, a numeric prediction of the judge's readings and its reasoning. Up to 1.5 times as many designs as needed are built and the most spread-out in physical quantities are kept.

Physics edits. The parent is the current best design with probability one third, otherwise an archive design. The target $\mathbf{g}$ is the normalized gradient of the model's predicted score with respect to the z-scored quantities. The Jacobian $J = \partial\phi/\partial\mathbf{u}$ is measured by central differences, setting each knob 0.2 of its range above and below the parent (for the 240-knob Go1, 2×240 variants in about 13 s, cached per parent, at most three new ones a round). The edit is

$$\Delta\mathbf{u} \propto \left(J^\top J + \epsilon I\right)^{-1} J^\top \mathbf{g}, \qquad \epsilon = 0.05, \tag{2}$$

where the damping keeps knobs that barely move any quantity from being pushed far. It is applied at four step sizes (0.03, 0.06, 0.10 and 0.15 of the knob range, root-mean-square) plus twelve perturbed copies; each candidate's quantities are predicted as $\phi + J\Delta\mathbf{u}$ and ranked by $\hat{\mu} + \kappa\hat{\sigma} - \beta\|\Delta\mathbf{u}\|^2$ ($\kappa = 1$, $\beta = 3$), and the best candidate that passes the novelty screen is proposed.

Search map. After the first round a goal score is fixed at the level about a third of the first-round designs reached; each region keeps a Beta$(1 + k, 1 + n - k)$ posterior over clearing it (k of n drawn

designs cleared; a design that cannot stand is a miss) and regions are chosen by Thompson sampling. The archive is described by the judge's first two graded readings; a design joins it when it is at least 0.12 (normalized) from every archived design or better than its nearest one.

Design screens. Searching and judging one hurdle design takes 100–200 s (about 2,650 episode equivalents); each screen takes well under a second. *Validity*: the design holds its home command for one episode without its trunk tilting past about 45° or sinking below half its height, without self-intersection, and, for a gripper, with a jaw that opens at least 5 mm wider and closes 5 mm narrower than the object within 1.5 s. *Novelty*: a candidate closer in z-scored quantities to a simulated design than the 10th percentile of nearest-neighbour distances (floored at 0.25, capped at 1.0) is skipped. *Predicted score*: once the model is trusted, the half of $2N$ candidates with the highest $\hat{\mu}$ go forward.

Planner, warm start and control screen. Commands are offsets from the home servo targets, parameterized by knots (App. A.2.3); a quadruped's 12 motors are grouped into four leg groups, other robots are commanded per motor. Each new design's planner starts from the current best design's executed commands on its best start, shifted by the difference in home commands and clipped to the new range. From its fourth re-plan, a design's search stops once its plan scores below the median of the other designs at that re-plan (with at least six recorded), and a design whose first start is outside the top 40% of the last 16 is not flown on further starts (the first four designs and a random 15% of stopped ones continue).

Choosing the best robot. On the hurdle the robot starts 0.60 ± 0.10 m behind the board; for bolted arms and hands only the planner's random numbers differ. Draws are extended, never redrawn, so all designs share starts. The held-out starts come from a separate random stream.

Learned design model. An ensemble of 16 ridge regressions ($\lambda = 1$) on bootstrap resamples of the run's designs predicts the score and up to three graded readings from the z-scored quantities; $\hat{\mu}$ is the fit on all designs and $\hat{\sigma}$ the ensemble spread. It is refitted with the z-scoring and the novelty distance after every round, uses no other runs, and is trusted once it has 12 designs and an out-of-fold Spearman $\rho \geq 0.2$ (five folds from ten designs on).

Printed parts and user edits. Loads are recorded from the replayed winning motion and the part is optimized in printed PLA as described in App. A.10. A user can constrain where the print may join the stock robot (for example, only at the foot over at least three quarters of its surface), how far new material may grow (two voxels) and which regions stay clear (30 mm around the knee), without rerunning the search; a second solver (CalculiX) re-checks the start and final parts.

A.4 Experimental Details

Evaluation protocol. All methods share the simulator (MuJoCo 3.11 (Todorov et al., 2012)), each task's design space, closed-loop CEM controller settings (App. A.2.3) and frozen judge. A design's score is 10 plus its graded reading if it succeeds and the fraction of the task completed otherwise (§2.3.2). During search every design is scored on one starting state, each round's three best on four, and the three best of the run on six; the final robot is the best of these and is re-judged on six held-out starts. We report the final robot's score, its gain over the stock robot, and the *pass rate*, the share of simulated designs that succeed. Intervals are bootstrap 95% intervals over three seeds.

Table 4 lists the parameters of every study in §3. All studies use failtask's frozen, unsigned gate, the start ladder of §3.1 (one start per design, four for each round's three best, six for the run's three best, six held-out starts for the final robot) and MuJoCo 3.11, except where noted.

Rewards. To compare designs under one objective, the hurdle, toss, bin pick and free throw use a fixed per-task reward file in L2D, VLMgineer, the black-box searches and the ablations; RoboMoRe and Eureka rewrite the reward as part of their method. On pen spin L2D uses one reward written by the language model, and the baselines the task's built-in reward.

Starting motion. In the language-driven comparison, the zero-shot motion seeds the planner on the hurdle, toss and pen spin. On bin pick and the free throw it displaced the tasks' scripted motions and nothing scored, so those runs were restarted after about four hours from the scripted motions, as for the baselines. The black-box searches and the ablations use no starting motion.

Hurdle consistency check. Seed 2 of L2D's language-free hurdle run was repeated from scratch after a thread race in concurrent simulation was fixed, for 460 designs. It agrees with the main run:

its best design scores 11.0, 67% of its designs succeed (against 67% for the main runs), and its three finalists score 10.96, 10.96 and 10.88 on six starts. The race had changed the stored physical quantities of 4 of about 1,200 designs and never a score.

Control fine-tuning. To test whether the free throw is limited by control, each seed's best L2D design was held fixed and its planner rerun on the same six held-out starts with 24 iterations and twice the samples (about 27,700 episodes per design), seeded from the task's scripted motion. The mean final score rose from 1.89 to 9.87, just below VLMgineer's 10.44; seeds 1 and 2 scored 11.00 on all six starts and seed 0 passed four of six. The baselines' designs were not given the longer search. On bin pick the same search changed the score by at most 0.12. On the hurdle, toss and pen spin, whose study controllers carry each design's found control from the incumbent, the longer search scored below the study finals whether it started from the written motion (hurdle 7.8, toss 10.5, pen 0.12) or from the design's own control (hurdle 12.4, toss 20.3, pen 0.13), against study finals of 15.5, 28.8 and 0.78; we therefore report the study finals there.

Table 2: **L2D against language-driven co-design at $3.33 of language model per seed.** Final: score on six held-out starts. Gain: over the stock robot. Pass: share of simulated designs that succeed. Designs: proposed per seed. Mean [95% interval] over seeds. Bold: best per task.

Task	Method	Final	Gain	Pass	Designs
Hurdle (Go1)	Eureka	10.82 [10.79, 10.83]	0.07 [0.04, 0.08]	0.79	74
	RoboMoRe	10.82 [10.79, 10.83]	0.07 [0.04, 0.08]	**0.95**	96
	VLMgineer	10.92 [10.88, 10.96]	0.17 [0.12, 0.21]	0.76	161
	L2D	**15.45 [14.92, 16.31]**	**4.70 [4.17, 5.56]**	0.78	**514**
Toss (UR5e)	Eureka	13.04 [12.58, 13.36]	4.06 [3.59, 4.37]	0.29	76
	RoboMoRe	15.48 [13.07, 17.88]	6.49 [4.08, 8.90]	0.54	96
	VLMgineer	19.77 [18.98, 21.26]	10.78 [10.00, 12.27]	**0.73**	149
	L2D	**28.82 [15.69, 42.58]**	**19.84 [6.71, 33.59]**	0.50	**660**
Bin pick (FR3)	Eureka	14.71 [14.68, 14.75]	1.09 [1.06, 1.14]	0.80	62
	RoboMoRe	**14.81 [14.74, 14.93]**	**1.20 [1.13, 1.31]**	0.88	67
	VLMgineer	14.64 [14.38, 14.78]	1.03 [0.77, 1.16]	0.80	192
	L2D	14.78 [14.75, 14.79]	1.16 [1.14, 1.18]	**0.93**	**1979**
Pen spin (hand)	Eureka	0.15 [0.07, 0.19]	0.03 [−0.05, 0.07]	0.00	56
	RoboMoRe	0.14	0.03	0.00	80
	VLMgineer	–	–	0.00	72
	L2D	**0.78 [0.21, 1.90]**	**0.66 [0.09, 1.78]**	**0.00**	**960**
Free throw (G1)	Eureka	3.10 [0.88, 4.25]	0.56 [−1.66, 1.70]	0.07	52
	RoboMoRe	7.63	5.09	0.03	72
	VLMgineer	**10.44 [9.32, 11.00]**	**7.90 [6.78, 8.46]**	**0.36**	160
	L2D	1.89 [0.71, 4.21]	−0.65 [−1.83, 1.67]	0.01	**813**

A.5 Additional Results

Language and the physics feedback. In an earlier version of the loop, on a Go2 wall-jump task with four seeds, the language model shown a learned map from design features to behaviour found higher jumps than the loop without language on every seed (+3.0 cm), while without the map its designs were worse on every seed (−0.8 cm).

Hurdle ablation. On the hurdle the best scores saturate, so the share of designs that clear is the useful reading. The warm start also shortens each design's search by about 20%. A pretrained world model used as the screen performs like the learned design model and is 17% cheaper per design, but neither ranks hurdle designs better than chance.

Transfer. On the Go2 pit leap, the seeded runs' best design was better after 10, 40 and 70 designs (by 0.45, 0.46 and 1.29). In every seeded run but two, the winning design did not descend from a

Table 3: **Ablation of the co-design loop** (mean ± s.d. over 3 seeds; 60 designs a cell on toss, 80 on the hurdle). *Toss* (UR5e+2F85, throw a ball as far as possible): distance of the best design re-judged on 6 starts, mean distance over every flown design (a failed throw counts 0 m), share of flown designs passing the throw gate, and seconds per design. *Hurdle* (Go1 over a 0.30 m rail): best re-judged score (≥10 = cleared; the vendor already clears on most starts, so this column saturates), share of the best design's starts cleared without touching the rail, share of flown designs clearing, and mean score of flown designs. Vendor: 4.5 m toss; hurdle score 9.10. †Hurdle wall-clock from seed 0 only, the one seed whose conditions all ran on the same machine.

	Toss				Hurdle				
Condition	Best (m)	Mean (m)	Pass	s/des.	Best	Clean	Clear	Mean	s/des.†
Full loop	14.2±4.0	5.1±3.4	0.57±0.21	422±11	**10.86**±0.07	**0.44**±0.28	0.39±0.09	5.67±0.81	129
− design screen	13.9±3.5	4.1±1.9	0.52±0.16	429±14	10.80±0.04	0.22±0.16	0.27±0.09	4.55±0.76	137
screen → world model	–	–	–	–	10.81±0.04	0.22±0.16	0.37±0.03	5.27±0.53	**107**
− control warm start	7.2±0.5	1.9±0.5	0.41±0.08	427±75	10.85±0.09	0.39±0.34	0.29±0.08	4.45±1.03	155
− diversity rule	**15.2**±3.1	5.5±3.3	0.59±0.25	413±18	**10.86**±0.09	**0.44**±0.34	0.40±0.03	5.59±0.48	125
− info chain (base)	8.5±1.8	2.6±0.4	0.54±0.06	377±55	**10.86**±0.05	**0.44**±0.21	0.46±0.03	6.19±0.29	148
info → language (Astra)	9.8±0.6	**6.6**±0.6	**0.83**±0.09	**335**±40	10.28±0.79	0.28±0.28	**0.56**±0.19	**7.27**±1.53	185

Table 4: **Studies and their parameters.**

Study	Tasks	Methods	Budget per seed	Other settings
Language-driven comparison (§4)	all five	L2D; RoboMoRe; VLMgineer; Eureka	$3.33 of GPT-6 Astra (low effort, capped output length); 3 seeds	L2D: 10 designs a round; baselines: 8 a round, ≤60 rounds, no filters; Lambda Cloud (L2D) and Google Cloud (baselines)
Black-box search (§4)	all five	L2D without language; random; BO; CMA-ES	1,000 designs; 3 seeds (L2D), 4–7 (baselines)	L2D: 12 designs a round, 25% fresh; BO: Matérn-5/2 GP, EI over 512 candidates; CMA-ES from the stock design, $\sigma = 1$ in calibrated widths; baselines judged in MuJoCo 3.3
Ablations (§4)	hurdle, toss	full loop without language and six variants	80 designs (hurdle), 60 (toss); 3 seeds	10 designs a round; language variant $7 of GPT-6 Astra; world-model screen: a pretrained quadruped capability model
Additional tasks (§4)	curling (Gen3), rocky slope (ANYmal C), hammer (Panda gripper)	L2D with language	30–130 designs; 3 seeds	$1.67 per seed; reward and motion written by the language model
Design transfer (§4)	Go2 long jump→pit leap; Go1 long jump→pit leap; toss→curling	from scratch vs. seeded with the source's two best designs	80 designs; 3 seeds	L2D with language; a language-free repeat also seeds the learned design model and region posteriors (running)

seeded one, so the benefit is a better starting point rather than inherited designs. Without language, seeding the long jump with pit-leap designs gave a design that crosses on some starts from the first proposal (against 48 proposals without seeding on one seed), now through designs descended from the seeded ones, and the prior-only arm improved more slowly. Seeding curling with toss designs gave weak evidence at best: counting a stone that leaves the ice as a miss, the seeded runs end ahead on two of three seeds (by 0.15 on average) and tie on the third. Without language, neither form of transfer had a reliable effect on toss to curling or on long jump to pit leap, where no design crossed the pit and final scores moved by at most 0.04. On a Go1, whose long-jump source never crossed its own pit, seeding gave no gain.

A.6 World Models as Design Screens, Proposers and Planners

Before settling on the learned design model of §2.3.2, we tried learned world models in every role a model can play in a co-design loop, in earlier versions of the loop and on quadruped jumping, hurdling and arm tasks. None survived into L2D.

Over the month before submission we ran 114 such experiments; Fig. 12 and Table 6 summarise the clearest.

Design ranker. A capability world model trained on 41,472 episodes (288 quadruped bodies, six tasks, 24 controls each) ranks fresh bodies at Spearman $+0.54$, but flying each candidate once ranks them better ($+0.63$), and on a held-out task the gap widens to -0.35. Inside a loop, where the

Table 5: **Design search on five tasks, no language anywhere.** Mean ± bootstrap 95% CI over seeds. *Gain* is the best design's score minus the run's own vendor design (the co-design loop and the three baselines ran different simulator versions, and on the hurdle different judges, so only gains and success rates are compared across them; †: the co-design loop's hurdle is a different course whose vendor already clears, not comparable). *Success*: share of evaluated designs that pass the task's frozen gate. *Budget*: designs evaluated, every draw counted; for the co-design loop on pen and the free throw, proposals (12 a round, standing or not), otherwise its flown designs (it redraws designs that do not stand). Budgets below 1000 are still running or were capped.

Task	Method	Runs	Budget/run	Gain over vendor ↑	Success (%) ↑	Runs w/ pass	First pass (#) ↓	s/design
Hurdle (Go1)	Random	4	1000	0.70 ± 0.14	0.0 ± 0.0	0/4	–	41
	BO	4	1000	2.99 ± 2.34	0.0 ± 0.0	1/4	940	59
	CMA-ES	4	1000	**5.50 ± 4.70**	0.8 ± 0.8	2/4	278	98
	Co-design loop†	3	1000	0.25 ± 0.00	67.6 ± 9.5	3/3	2	328
Pen spin (hand)	Random	4	1000	0.03 ± 0.05	0.0 ± 0.0	0/4	–	1
	BO	4	1000	0.14 ± 0.17	0.0 ± 0.0	0/4	–	45
	CMA-ES	4	1000	**0.32 ± 0.01**	0.0 ± 0.0	0/4	–	120
	Co-design loop	2	996	0.24 ± 0.01	0.0 ± 0.0	0/2	–	56
	Co-design loop + language	5	9–92	0.14 ± 0.03	0.0 ± 0.0	0/5	–	605
Toss (UR5e)	Random	4	1000	3.59 ± 1.84	4.3 ± 0.1	4/4	10	58
	BO	4	1000	14.21 ± 2.59	31.1 ± 8.9	4/4	8	87
	CMA-ES	4	1000	26.86 ± 2.92	52.7 ± 2.9	4/4	2	135
	Co-design loop	3	1000	**40.28 ± 21.56**	80.0 ± 9.9	3/3	3	267
	Co-design loop + language	3	128–144	38.28 ± 9.84	65.7 ± 7.2	3/3	3	425
Bin pick (FR3 gripper)	Random	5	370–1000	1.59 ± 0.04	26.1 ± 1.0	5/5	3	145
	BO	6	228–1000	**1.70 ± 0.08**	69.6 ± 6.8	6/6	5	111
	CMA-ES	6	519–1000	1.66 ± 0.04	62.0 ± 6.5	6/6	2	100
	Co-design loop	3	1000	1.31 ± 0.16	92.6 ± 5.9	3/3	2	70
	Co-design loop + language	3	192–382	1.12 ± 0.05	86.5 ± 3.7	3/3	2	129
Free throw (G1)	Random	14	178–1000	-0.87 ± 0.01	0.0 ± 0.0	0/14	–	251
	BO	8	325–1000	8.75 ± 2.75	24.1 ± 11.4	7/8	25	198
	CMA-ES	11	300–1000	**10.13 ± 0.00**	31.4 ± 6.0	11/11	13	148
	Co-design loop	3	132–600	8.45 ± 0.00	2.9 ± 2.7	3/3	36	143
	Co-design loop + language	4	28–39	8.45 ± 0.00	41.6 ± 21.4	4/4	8	641

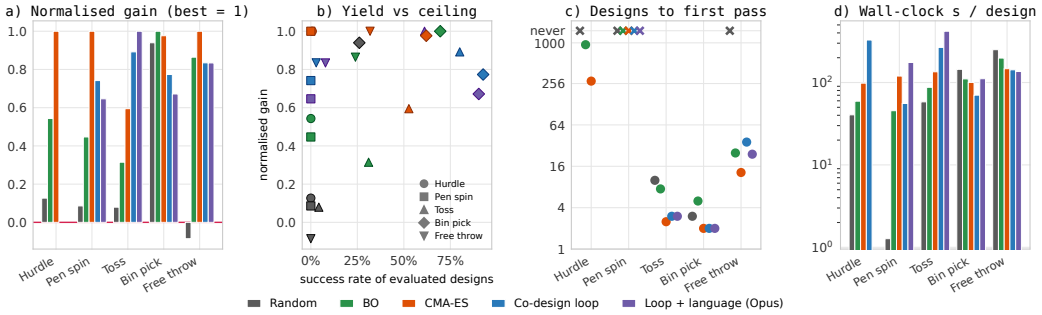


Figure 11: **Design search without language, per task.** (a) Normalized gain, (b) success rate against gain, (c) designs to the first pass and (d) wall-clock seconds per design (log scale). The loop's hurdle ran a different course from the baselines' and is not comparable.

candidates differ only in the parts the prompt allows, its ranking falls to chance ($\rho = +0.03$ with 14 calf knobs live against $+0.15$ with the whole body; -0.05 in the loop). In the hurdle ablation (§4), the same pretrained model used as the design screen was 17% cheaper per design than the learned design model but, like it, ranked designs no better than chance.

Rejector and proposer. A world model refitted on the base loop's own 1,583 hurdle designs rejected falling designs only weakly (AUC 0.63), and the designs it proposed lowered the mean score by 0.034 $[-0.085, +0.018]$ with twice as many falls. On a wall jump, a task-retrained model that both proposed and screened made the loop worse: 0.12 of its designs cleared the wall, the same as the stock robot, against 0.62 without the model, because it screened with the score it proposed by, and all 27 designs it forwarded were its own.

Design gradient. Stepping along a world model's gradient with respect to the design lost performance in simulation (-0.38 over eight arms) where ridge and random steps of the same size gained $+1.12$, even though its predicted gains correlated best with the real ones ($+0.71$). A correct slope is a much stronger demand than a correct ranking.

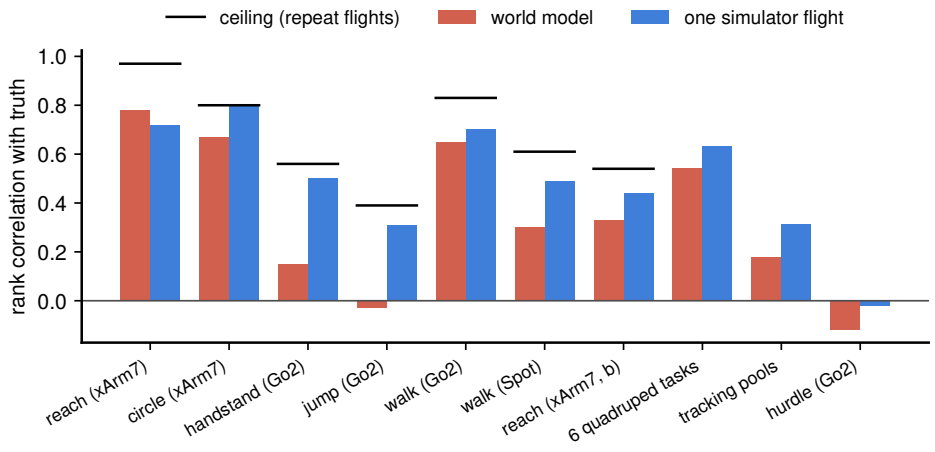


Figure 12: **Ranking designs: world model against one simulator flight.** Spearman correlation with the true ranking of the same designs, per study (bars: world model and a single simulated flight of each design; lines: the ceiling from repeated flights, where measured). The model wins 1 of 10, on a bolted arm, and loses every contact-rich task.

Table 6: **World models in each role** (mean with 95% interval where measured).

Role	Setting	World model	Comparison
Design ranker	six quadruped tasks, offline	$\rho = +0.54$	one flight $+0.63$ (-0.09 [-0.19, -0.00])
Design screen	hurdle ablation (§4)	$\rho = -0.12$, 37% of designs clear	ridge screen $\rho = -0.13$, 39%
Design proposer	Go2 hurdle loop	mean score -0.034 [-0.085, $+0.018$]; 52% of designs fall	no model; 33% fall
Proposer and screen	Go1 wall jump	12% of designs clear (as stock)	62% without the model
Design gradient	eight arms	gain -0.38	ridge or random step $+1.12$
Controller	six quadruped tasks	-0.18 [-0.28, -0.08]	replaying the written motion
Warm start	25 seeds, equal budget	-0.020 [-0.064, $+0.023$]	cold CEM
Warm start into CEM	Go1 wall jump	$+0.216$ m [$+0.016$, $+0.416$] apex	CEM from the written motion

Planner and warm start. As a motion planner, a world model ranked candidate motions correctly only while the robot stood near its start and backwards once it was moving (Spearman -0.32 to -0.64), as its predicted position drifted by about 1 m over a 0.96 s horizon. As a warm start, iLQR discarded its plan (the result was identical to iLQR without a model), and it lost to any control that had already been searched (6 of 6). A dynamics model carried across designs to accelerate policy search gained nothing (-0.025 ± 0.041). Warm starts did help at low search budgets, on arms and on new robots, and fed into a CEM they raised the wall jump's apex (Table 6), but at equal budget they did not beat a cold search or a carried control, which is why L2D warm-starts from the best searched motion.

Why they fail. The studies share one cause. A design's score depends on the motion found for it, and a model trained on drawn or earlier controls cannot see what a search will find for a new body, so it ranks bodies under the wrong controller, drifts over the horizon a dynamic task needs, and has the least signal in the narrow region a loop searches. L2D therefore uses a model only where being wrong is cheap: the learned design model orders candidates for the screen and suggests directions for the physics edits, and every design that counts is simulated and judged.

A.7 Round-Trip Accuracy of the Design Space

Reconstruction is measured by encoding a robot, rebuilding it, and simulating the original and the rebuild for 60 steps under identical commands, in free flight and on the ground. A rebuild is *exact* if it has the same numbers of joints, velocities and actuators and every body stays within 0.01 mm (free flight) and 0.1 mm (ground). *Within floor* means within 1.5× the simulator's own numerical floor: the difference between two rebuilds that differ only in each mesh's local frame (Table 7).

Table 7: **What the design space represents, and how exactly it rebuilds robots.**

Geometry	primitives, meshes as the simulator's collision hull, and generated contact parts as their shape parameters; inertia stated exactly
Closed chains	equality constraints and tendons (e.g. four-bar grippers)
Actuators	full gear vector, gains, activation, in control order; on a joint, tendon or body
Round-trip accuracy	31/31 corpus robots exact; 520/532 edited designs exact, 532/532 within floor; 86/103 unseen Menagerie models exact, 89/103 within floor
Generated contact shapes	270/270 exact; a cupped foot is 101 parameters instead of 38,497 hull vertices
Robot families	97 starters in nine families (1–43 driven joints, 0.3–109 kg) plus 43 measured edits

A.8 Co-Design Loop Pseudocode

Algorithm 1 gives the full loop of §2.3 in its default configuration.

Algorithm 1 L2D co-design loop (default configuration)

Require: sentence ℓ, scene $\mathcal{S}$ with vendor M_0, judge J; rounds R, designs per round N

Freeze gate $\mathcal{G}$ (spec + hash of judge source); write reward r_ℓ, test it against standing still, freeze it
Optionally write motion seed $\mathbf{a}^{\text{seed}}$; fly vendor θ_0; fit z-scaler and novelty metric on $\phi(\theta_0)$
Calibrate each design region's width until $\geq 75\%$ of draws stand
for $t = 0, \ldots, R-1$ **do**
  if surrogate not ready **or** $t \bmod k_{\text{reseed}} = 0$ **then** ▷ language seat
    $\mathcal{C}_{\text{edit}} \leftarrow$ LanguageSeat(ℓ, tables_t); fill shortfall with InfoChain
  else ▷ information chain
    $\mathcal{C}_{\text{edit}} \leftarrow$ InfoChain$(\theta^{\text{inc}}, \text{elites}, \hat{f}_t)$
  end if
  $\mathcal{C}_{\text{fresh}} \leftarrow$ Thompson-sampled region draws ($p_{\text{fresh}} = 0.25$ of proposals)
  Keep candidates that stand and pass the novelty rule; if $\hat{f}_t$ is ready, over-propose $2N$ and keep the top-N by $\hat{\mu}$
  for each admitted θ **do**
    $\mathbf{a} \leftarrow$ CEM-MPC on r_ℓ, warm-started from $\text{transfer}(\mathbf{a}^{\text{inc}} \to \theta)$, common random numbers
    Re-fly $\mathbf{a}$ on $M(\theta)$ from the same s_0; score $\mathcal{G}(\tau)$; offer θ to the archive
  end for
  Promote the round's top 3 to n_{mid}=6 starts; score the language model's predictions
  Refit z-scaler, novelty radius, ridge ensemble $\hat{f}_{t+1}$; update region posteriors
end for
Re-judge the top 3 on 12 starts; fly the pick and the vendor on held-out starts; report pass rate and Wilson CI

A.9 Comparison with Prior Co-Design

Table 8: **Comparison with prior co-design.** ✓: yes; ∼: partially; ✗: no.

Method	Cross-family	Topology edits	Generated contact solids	Language task	Physics-grounded proposer	Judge ≠ reward	Training-free control
NGE (Wang et al., 2019)	✗	✓	✗	✗	✗	✗	✗
Transform2Act (Yuan et al., 2022)	✗	✓	✗	✗	✗	✗	✗
BodyGen (Lu et al., 2025)	✗	✓	✗	✗	✗	✗	✗
Transformer Transformer (Ha et al., 2026)	✓	∼	✗	✗	∼	✗	∼
Text2Robot (Ringel et al., 2025)	✗	✗	∼	✓	✗	✗	✗
LASeR (Song et al., 2025)	✗	✓	✗	∼	✗	✗	✗
RoboMoRe (Fang et al., 2025)	✗	∼	✗	✓	✗	✗	✗
VLMgineer (Gao et al., 2026)	✗	✗	✓	✓	✗	∼	✓
Debate2Create (Qiu & Cygan, 2026)	✗	∼	✗	✓	✗	∼	✗
L2D (ours)	✓	✓	✓	✓	✓	✓	✓

A.10 From Design to a Printable, Lightweight Part

The final stage turns the printed add-on of a winning additive design into a part that is as light as the task allows and strong enough for it.

Loads. The chosen motion is re-flown in the loop's own engine and controller (closed-loop CEM-MPC over the stock position servos). For each printed part and physics step we record, in the part's own frame, the joint reaction at the knee, every contact point and force on the part, and the body's rigid acceleration field. The load cases are the frames of peak contact force and peak knee torque of each leg.

Finite-element model. One regular voxel grid of pitch h (1.5–3 mm) covers the part, padded so that material can also be added. Each voxel is void, stock (inside the vendor link; never removed) or print, and every solid voxel is a trilinear hexahedron. Printed PLA is linear elastic ($E = 3.5\,\text{GPa}$, $\nu = 0.36$, $\rho = 1240\,\text{kg/m}^3$). The stock aluminium shell is given the homogenised modulus of a shell of the same mass. The print is fused to the stock link only through the foot: elsewhere it has its own nodes, so a sleeve rides the link without carrying load into it. The knee is clamped; contact forces are spread over nearby surface nodes; gravity and inertia enter as the body force $-\rho\,\mathbf{a}(\mathbf{x})$, with $\mathbf{a}(\mathbf{x}) = \mathbf{a}_0 + \boldsymbol{\alpha} \times \mathbf{x} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x})$. The same discrete problem is solved by CalculiX, FEniCSx, scikit-fem, NGSolve and a NumPy assembly, which agree to solver precision; the fastest runs the optimisation and CalculiX re-solves the initial and final parts as a check.

Optimisation. With $\rho_e \in \{0, 1\}$ the print occupancy of voxel e, we seek

$$\min_{\rho}\ m(\rho) \quad \text{s.t.} \quad \max_{c}\ \sigma^{\text{vM}}_{e,c} \le \sigma_{\text{allow}}\ \ \forall e, \qquad \delta_{\text{foot}} \le 1.5\,\delta^{0}_{\text{foot}}, \qquad \text{cover} \ge 0.75, \tag{3}$$

where c ranges over load cases, $\sigma_{\text{allow}} = 20\,\text{MPa}$ (half of printed PLA's tensile strength along the layers), and cover is the fraction of the stock foot's exposed faces fused to print. We use bidirectional evolutionary structural optimisation (BESO): each iteration takes the von Mises envelope over load cases, smoothed over each voxel's $3 \times 3 \times 3$ neighbourhood, removes the least-stressed 4% of removable print, and adds print on the void neighbours of voxels above a fraction of σ_{allow}, within two voxels of the original part and at least 30 mm from the knee. Print near every contact patch is frozen, and material that no longer carries load to the knee is dropped. The returned part is the last iterate meeting all three constraints; it is exported as an STL in the link's frame, printed, and fitted over the stock link.

Procedure. (1) Re-fly the winning motion and extract the load cases. (2) Voxelise the part at 2 mm and label each voxel void, stock or print. (3) If the print covers less than three quarters of the stock foot's exposed faces, first grow it over the foot until it does. (4) Iterate BESO for up to 60 steps, solving every load case at each step and recording mass, peak stress, foot deflection and cover. (5) Return the lightest iterate that meets every constraint, re-solve the start and the result in CalculiX, and export the STL.

Table 9: **Constraints and settings of the part optimisation.** The last four rows are the user's edits (§2.4); they can be changed without rerunning the co-design search.

Constraint or setting	Value	Why
Peak von Mises stress	≤ 20 MPa	half of printed PLA's tensile strength along the layers; evaluated outside the contact patches, where point loads are singular
Foot deflection	$\le 1.5\times$ the start	the optimised part must stay as stiff as the design that was flown
Stock material	never removed	the vendor link is not modified
Removal and addition rate	4% removed, add above $0.9\,\sigma_{\text{allow}}$	BESO step sizes; stresses smoothed over $3 \times 3 \times 3$ voxels to avoid checkerboards
Contact patches	frozen within 6 mm	the surfaces the motion touches carry its loads
Knee clamp	18 mm radius	the load path ends at the knee joint
Load path	connected to the knee	material that carries no load is dropped
Attachment (weld)	foot only, cover ≥ 0.75	the print may join the stock link only at the foot, over most of its surface
Growth	within 2 voxels	the part stays printable around the part it came from
Knee clearance	30 mm	nothing is added near the knee, which must stay free to move
Voxel pitch	2 mm (1.5–3 mm)	resolution against solve time