DIFFERENTIABLE EVOLUTION OF AGILE AGENTS

Anonymous authors
Paper under double-blind review

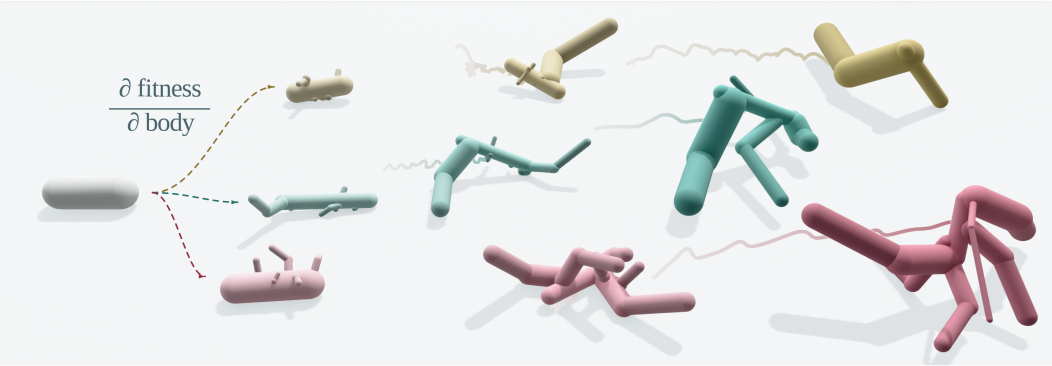

Figure 1: Articulated rigid-bodied robots evolve from a common ancestor (gray link, *left*) into distinct species (*right*) following the gradient of fitness with respect to the robot's body plan. Videos can be found at project page.

ABSTRACT

It is usually much easier to compute meaningful policy gradients than meaningful design gradients, especially in jointed rigid bodies that move quickly and gracefully on land. Whereas soft bodies without bones and joints naturally lend themselves to continuous modeling, they are generally slower, weaker, and less interesting in terrestrial environments. Here we demonstrate end-to-end differentiable evolution of articulated agents capable of agile legged locomotion. Starting from a primordial capsule, jointed links emerge and grow and move along the body as others shrink and disappear, across evolutionary time, depending on derivatives of fitness with respect to design variables. These variables were sampled from a distribution to produce a population of designs, which were controlled by a shared universal policy. Gradients were averaged across individuals in the population and across truncated learning windows inside each forward simulation to update the design distribution and universal controller concurrently. This enables the sample-efficient discovery and gradient-based optimization of novel robots with exciting athletic behaviors.

1 INTRODUCTION

Learning and evolution can be seen as formally equivalent processes of incremental adaptation operating on different timescales (Pigliucci, 2008; Friston, 2010; Watson & Szathmáry, 2016). The essential difference is that while learning occurs within a given species, evolution transmutates one species into another. Thus, fins become limbs and feet become hands with which increasingly sophisticated behaviors can be learned. In robots, evolution and learning are usually conducted in simulation (Lipson & Pollack, 2000; Yu et al., 2026) and under certain conditions, gradients of behavior can be computed with respect to the robot's physical layout (Matthews et al., 2023).

To date however, the conditions that enabled differentiable evolution have required robots to be entirely soft, jointless, volumetrically actuated bodies (Ma et al., 2021; Matthews et al., 2023; Cochevelou et al., 2023; Kobayashi et al., 2024). This allowed every part of robot's body to continuously increase and decrease in physicality (mass, elasticity) and actuation amplitude, while also

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.

smoothing the robot's dynamics such that useful design gradients could be backpropagated across long periods of time. But these simplifying assumptions inhibited the robot's movement on land. In the extreme case, gradient computation required the robot to be suspended in a fluid (Ma et al., 2021). Although others (Matthews et al., 2023; Cochevelou et al., 2023; Kobayashi et al., 2024) exhibited terrestrial locomotion, their robots were in many ways like a fish out of water, bouncing awkwardly along the surface. Some gradient-based approaches have been reported for optimizing the orientation or length of a rigid bodied robot's limbs (Ha et al., 2017; Spielberg et al., 2017) but the robot lacked agility and its overall structure was presupposed and held fixed—new species could not emerge. Similarly, Xu et al. (2021) optimized the geometry of links within an immotile serial chain manipulator using gradient information from differentiable simulation. Strgar et al. (2024) used simulation gradients to learn graceful locomotion in evolving robots, but evolution itself was not differentiable: mutations randomly added and removed body parts. Simulation gradients have also been used to accelerate policy learning but not morphological evolution in agile articulated agents (Xu et al., 2022; Georgiev et al., 2024; Xing et al., 2025).

Without access to first-order analytic gradients of task performance with respect to morphology, the utility of potential design changes must be estimated from sampled task returns. Schaff et al. (2019) did so by optimizing a distribution over continuous design parameters with a score-function estimator, whereas Yuan et al. (2022) modeled morphological transformations as actions to be learned jointly with control policies. Lu et al. (2025) identified unbalanced reward signals between the design and control stages as a source of optimization inefficiency and introduced temporal credit assignment to rebalance this zeroth-order formulation.

Here we introduce the differentiable evolution of articulated robots capable of moving quickly across terrestrial environments. We present **FORM** (First-Order Robot Morphogenesis), a method for jointly co-optimizing morphology and control by differentiating task objectives through rigid-body dynamics and contact. To do so, FORM leverages short-horizon differentiation and represents structural variation through continuous variables over a candidate kinematic supergraph, thereby preserving pathwise morphology gradients without presupposing the agent's final topology.

2 METHODS

2.1 FORM: FIRST-ORDER ROBOT MORPHOGENESIS

FORM jointly co-optimizes the topology, geometry and closed-loop behaviors of kinematic trees using gradient information from differentiable simulation. Let s_t, a_t, and m denote the state, action, and morphology, respectively. The policy samples $a_t \sim \pi_\theta(\cdot \mid s_t, m)$ and the simulator advances the state as $s_{t+1} = f(s_t, a_t, m)$. Morphologies were drawn from a normal distribution (i.e. of genotypes) centered at $\mu \in \mathbb{R}^d$, mapped to physical parameters (i.e. phenotypes) by $m = \mathcal{T}(\mu)$, and optimized by maximizing the expected discounted task return:

$$\max_{\theta,\mu} J(\theta, \mathcal{T}(\mu)), \qquad J(\theta, m) = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right],$$

where T is the episode length and γ is the discount factor.

Local morphology sampling. In each generation, k, of evolution a fraction, c, of the B parallel environments contain robots with the central form $\mathcal{T}(\mu_k)$ while the remainder contain robots with morphologies that are determined by antithetic Gaussian sampling. Specifically, for each $\epsilon_i \sim \mathcal{N}(0, I)$, we instantiate the pair:

$$\mathcal{T}(\mu_k + \sigma \odot \epsilon_i), \qquad \mathcal{T}(\mu_k - \sigma \odot \epsilon_i).$$

Here σ specifies the sampling scales in raw design space. The resulting population $\{m_i\}_{i=1}^B$ consists of center copies and antithetically sampled morphologies and is fixed throughout the generation. This population approximates the mixed objective

$$\mathcal{J}(\theta, \mu; c) = cJ(\theta, \mathcal{T}(\mu)) + \frac{1-c}{2}\,\mathbb{E}_\epsilon\left[J(\theta, \mathcal{T}(\mu + \sigma \odot \epsilon)) + J(\theta, \mathcal{T}(\mu - \sigma \odot \epsilon))\right].$$

Sampling exposes the controller and critics to a neighborhood of designs and provides local design signals when the center has weak or uninformative contacts (Suh et al., 2022). A shared,

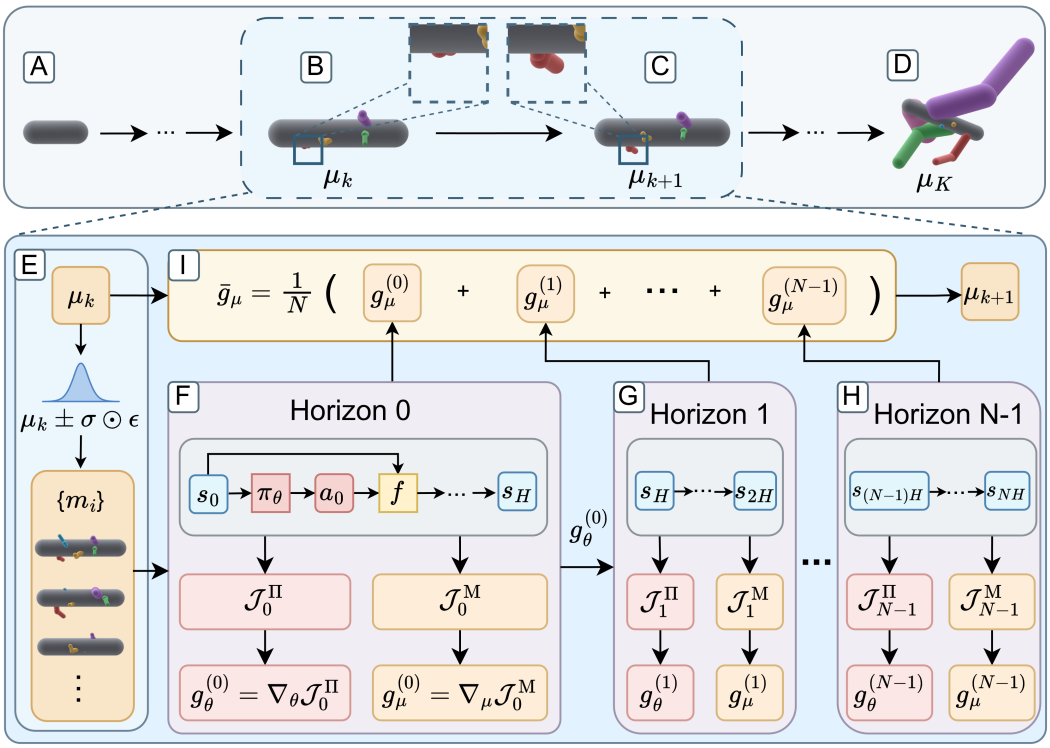

Figure 2: **Differentiable co-design of kinematic structure and neural control.** Robots begin as an immotile link (**A**) and pass through intermediate forms (**B** and **C**) as they evolve into a final, optimized morphology (**D**). In each generation of evolution, a population of robots is generated with morphologies sampled from a distribution centered about μ_k (**E**). The robots behave across N short-horizon chunks of simulation (**F–H**) using a shared, morphology-conditioned universal controller. Policy gradients are applied between consecutive horizons while morphology gradients are accumulated across horizons (**I**) and used to update the distribution from μ_k to μ_{k+1}.

morphology-conditioned "universal" controller transfers control experience across these designs (Yu et al., 2017; Gupta et al., 2022; Huang et al., 2020; Yuan et al., 2022; Li et al., 2025). Adjusting c balances direct training of the central form with exploration of nearby morphologies.

Short-horizon objectives. Following SHAC (Xu et al., 2022) and related first-order gradient-based methods (Georgiev et al., 2024; Xing et al., 2025), we split each episode into $N = T/H$ horizons of H steps. For horizon $n \in \{0, \ldots, N-1\}$, let $t_n = nH$. A terminal value estimate provides the continuation estimation of future objective beyond the simulated horizon. We use separate objectives for task-return morphology optimization and entropy-regularized control policy optimization:

$$\mathcal{J}_n^{\mathrm{M}} = \frac{1}{B}\sum_{i=1}^{B}\left[\sum_{h=0}^{H-1}\gamma^h r_{t_n+h,i} + \gamma^H V^{\mathrm{M}}(s_{t_n+H,i}, m_i)\right],$$

$$\mathcal{J}_n^{\Pi} = \frac{1}{BH}\sum_{i=1}^{B}\left[\sum_{h=0}^{H-1}\gamma^h (r_{t_n+h,i} + \alpha\widehat{h}_{t_n+h,i}) + \gamma^H V^{\Pi}(s_{t_n+H,i}, m_i)\right].$$

Here V^{M} estimates task return, while V^{Π} estimates entropy-augmented return. Following Xing et al. (2025), $\widehat{h}_t$ is a scaled policy entropy; the temperature α is adapted using the unscaled entropy estimate and target entropy. Both critic families are trained using TD(λ) targets (Sutton, 1988), with their respective reward signals.

Algorithm 1 FORM: First-Order Robot Morphogenesis

Input: Initial design center μ_0, policy parameters θ, and two critic families
Output: Final design center μ_K and policy parameters θ

```
Warm up the policy and critics with μ_0 fixed
for k = 0, ..., K − 1 do
    Choose the center fraction and sample the population {m_i}_{i=1}^B
    Reset the environments and initialize the morphology-gradient accumulator G_μ ← 0
    for n = 0, ..., N − 1 do
        Collect an H-step differentiable rollout with the population fixed
        Compute g_θ^(n) and g_μ^(n)
        θ ← Update_θ(θ, g_θ^(n))
        Fit both critic families and update the morphology-critic targets
        G_μ ← G_μ + g_μ^(n)
    end for
    ḡ_μ ← G_μ / N
    μ_{k+1} ← Update_μ(μ_k, ḡ_μ)
end for
```

Policy and morphology updates. The two objectives yield the first-order gradients:

$$g_\theta^{(n)} = \nabla_\theta \mathcal{J}_n^{\Pi}, \qquad g_\mu^{(n)} = \nabla_\mu \mathcal{J}_n^{\mathrm{M}}, \qquad \bar{g}_\mu = \frac{1}{N} \sum_{n=0}^{N-1} g_\mu^{(n)}.$$

We update the policy after every horizon and the design center once per generation using $\bar{g}_\mu$. Thus, the averaged morphology gradients aggregate morphology-relevant information from the successive controllers encountered within each generation. When computing $g_\mu^{(n)}$, policy and critic parameters are treated as constants, while derivatives through their state and morphology inputs are retained. When computing $g_\theta^{(n)}$, morphology and critic weights are treated as constants as well. At each horizon boundary, we detach the simulated state to truncate temporal gradients and reapply the differentiable morphology model to rebuild the gradient path from μ to the physical parameters for the next horizon. Algorithm 1 summarizes the resulting update cycle from μ_k to μ_{k+1}. See Fig. 2 for a visual abstract.

2.2 Implementation Deatils

We use $B = 64$ parallel environments, $T = 320$ control steps per generation, and $H = 8$ steps per horizon. Training begins with 16 warm-up episodes, during which the center is fixed and neighboring morphologies are resampled each episode, followed by $K = 256$ generations of morphological evolution. The sampling scales σ remain fixed during the mixed-population phase, and each sampled population is held for one generation. Policy optimization follows Soft Analytic Policy Optimization (SAPO; Xing et al. (2025)); morphology updates use Adam with a learning rate that decreases linearly from 0.15 to zero.

Center adjustment. We adapt the center fraction using the mean intragenerational return R_{center} of the center environments. At the end of each generation, we update an exponential moving average of competence, $\bar{q}$, and map it to the center-fraction range:

$$\bar{q} \leftarrow 0.9\bar{q} + 0.1 \,\mathrm{clip}\left(\frac{R_{\mathrm{center}}}{R_{\mathrm{ref}}}, 0, 1\right), \qquad c = 0.125 + 0.625\bar{q},$$

with $R_{\mathrm{ref}} = 1.0$. Higher competence allocates more environments to the center. For the final 64 design updates, all environments use the center morphology while both policy and morphology optimization continue.

2.2.1 Model Structure

We parameterize π_θ as a tanh-squashed Gaussian policy and maintain two LayerNorm twin-critic families, one for the entropy-augmented policy return and one for the task-only morphology return.

As in SAPO (Xing et al., 2025), the policy objective uses the mean of its twin-critic predictions, while its TD targets use their minimum. For morphology optimization, both the objective and its TD targets use the minimum of two Polyak-averaged morphology-critic target predictions. Network architectures and optimization hyperparameters are provided in Table 2 and 3.

2.2.2 Simulator Environment

We implement articulated dynamics in Newton (The Newton Contributors, 2025), which is built on Warp (Macklin, 2022). A custom PyTorch autograd function connects policy and morphology tensors to Warp's simulation tape, inspired by the PyTorch–Warp interface implemented by Xing et al. (2025). Genotype-to-phenotype matrix products and their backward passes use IEEE FP32. Capsule-endpoint contacts use a fixed slot structure, with contact geometry recomputed from the current state and morphology at every simulation step. We use a local implicit tangential-friction update to handle small effective contact masses; further simulator details are provided in Appendix C.

2.3 Robot Design

We represent robot morphology using a capsule-shaped root link and a set of candidate articulated appendages. The root length and radius, (L_c, r_c), together with eight parameters per appendage, $(L_i, r_i, \zeta_i, \phi_i, p_i, \beta_{m,i}, \beta_{c,i}, \eta_i)$, define morphology m. These variables specify appendage size, attachment location, presence, deployment orientation, and hip-axis orientation (see Table 1).

Presence p_i provides a continuous relaxation within a fixed kinematic supergraph. It scales motor torque capacity, masks joint-position and joint-velocity observations, and controls geometry through

$$g(p) = 0.25 + 0.75p, \qquad w(p) = \frac{p^2}{p^2 + 0.125^2}.$$

The geometry scale $g(p)$ multiplies each appendage's length and radius, preserving a nonzero minimum size. As the docking weight w decreases, the hip moves toward the root center and the rest direction of the proximal segment rotates toward the root link's $+x$-axis. Low-presence links can subsequently be pruned, as evaluated in the experiments.

The map $\mathcal{T}$ transforms each bounded design variable $x \in [l_x, u_x]$ as

$$x = l_x + (u_x - l_x)\,\mathrm{sigm}(\mu_x),$$

where $\mathrm{sigm}(z) = (1+e^{-z})^{-1}$ denotes the logistic sigmoid. Attachment azimuths are instead treated as periodic variables. The presence is decoded as $p = \mathrm{sigm}(\mu_p)$, with its logit restricted to $[-3, 3]$. Appendices A and B give the complete geometric construction and actuator model.

3 Results

3.1 Experimental setup

FORM operates on a chosen candidate kinematic supergraph. In our experiments, the supergraph consists of a capsule root and can express up to six articulated appendages, yielding a 50-dimensional design space. At initialization, the candidate links are small and fully contained within the root, so the robot has no appendages. For each seed, the attachment and orientation variables $(\zeta_i, \phi_i, \beta_{m,i}, \beta_{c,i}, \eta_i)$ are sampled uniformly from their admissible ranges, rejecting configurations

Table 1: Morphology parameters

Symbol	Meaning	Symbol	Meaning
L_c	Root length	ζ_i	Meridional attachment coordinate
r_c	Root radius	ϕ_i	Attachment azimuth
L_i	Nominal appendage length	$\beta_{m,i}$	Meridional tilt
r_i	Nominal appendage radius	$\beta_{c,i}$	Circumferential tilt
p_i	Appendage presence	η_i	Hip-axis orientation

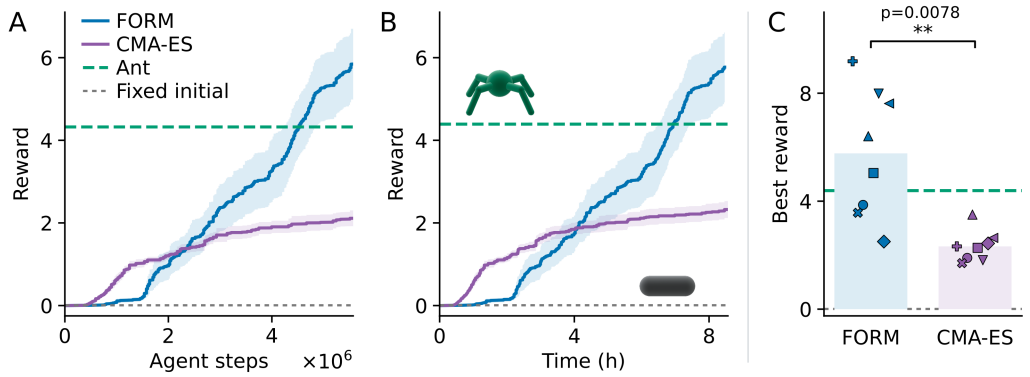

Figure 3: **FORM increases the evolvability of agile legged robots.** We compared FORM across eight independent evolutionary trials against an otherwise equivalent approach that approximates, rather than computes, design gradients, and visualized this as a function of agent steps (**A**) and wall-clock time (**B**). Cumulative max reward is plotted; shaded regions indicate standard error of the mean. The green dashed line indicates the best performance achieved by a popular quadrupedal design in the same task environment. The reward of the best designs found in each trial are shown (in **C**). FORM achieved the higher reward in every all eight paired trials with matched initializations ($p < 0.01$; two-sided Wilcoxon matched-pairs signed-rank test).

in which any link protrudes beyond the root; size and presence variables use their fixed initial values. Thus, all runs begin with the same capsule morphology but different internal candidate-link configurations.

The locomotion task uses no auxiliary shaping terms: the reward is only forward displacement along the global x-axis,

$$r_t = x_{t+1} - x_t.$$

Episodes last 320 control steps. At reset, all joint offsets from the rest configuration and generalized velocities are zero, and the root height is determined by the static-preload initialization described in Appendix C. The policy receives a 91-dimensional observation,

$$o_t = [h_t,\ (q_t^{\mathrm{root}})^\top,\ (\nu_t^{\mathrm{root}})^\top,\ (\tilde{p} \odot \dot{q}_t)^\top,\ (\tilde{p} \odot q_t)^\top,\ c(m)^\top]^\top.$$

This observation comprises root height (1), root orientation (4), root generalized velocity (6), presence-weighted joint velocities (12) and positions (12), and a 56-dimensional morphology encoding. Here $\tilde{p} = [p_1, \ldots, p_6, p_1, \ldots, p_6]^\top$ denotes the joint-presence mask. The morphology encoding $c(m)$ is obtained from the 50-dimensional design by representing each of the six attachment azimuths as a sine–cosine pair; presence is encoded as $2p_i - 1$, and the remaining coordinates are normalized to their design ranges. Only the 35 dynamic-state features use running-statistic normalization.

We compare FORM with CMA-ES over eight matched initialization seeds using the same morphology parameterization. After the same 16-episode warmup, each CMA-ES generation uses 56 candidate morphologies and 8 center copies; only candidate returns update the CMA-ES distribution, while all 64 rollouts train the controller and critics. For the agent-step comparison, both methods are truncated at 5.57 million steps; for the wall-clock comparison, CMA-ES runs for 378 generations to match FORM's training time. Other CMA-ES settings are given in Appendix E. Fixed-morphology references are the initial design and the fixed Ant-inspired morphology, trained with four controller seeds (Fig. 8; Appendix E). For each seed, we compute the best-so-far return by taking the cumulative maximum across its episodes. We then report the mean and standard error of the best-so-far return across seeds.

3.2 Locomotion and Morphological Development

Figure 3(a,b) shows that CMA-ES improves more rapidly initially, whereas FORM continues improving after CMA-ES begins to plateau. At 5.57 million agent steps, FORM reaches a mean best-so-far return of 5.85 ± 0.86, compared with 2.11 ± 0.19 for CMA-ES (mean $\pm$ s.e.). This advantage is also maintained under the matched wall-clock budget. FORM achieves the higher return in all



Figure 4: **Eight independent phylogenies.** Each row shows a different evolutionary trial, each starting from the same immotile link (*left*) and ending with a unique design (*right*).

eight matched runs (Fig. 3c; two-sided exact paired Wilcoxon signed-rank test, $p = 0.0078$). Five of the eight FORM runs also exceed the fixed Ant reference. In contrast, training the controller with the initial morphology fixed yields negligible forward movement over the training budget.

Figure 4 traces morphological development from the same immotile link. Across runs, morphological appendages emerge as candidate articulated structures grow beyond the root, yielding distinct articulated bodies. The rightmost snapshots show each run's best checkpoint; matching markers identify the corresponding runs in Figure 3(c). These trajectories show that the same candidate kinematic supergraph can yield diverse agile morphologies under the same forward-displacement objective.

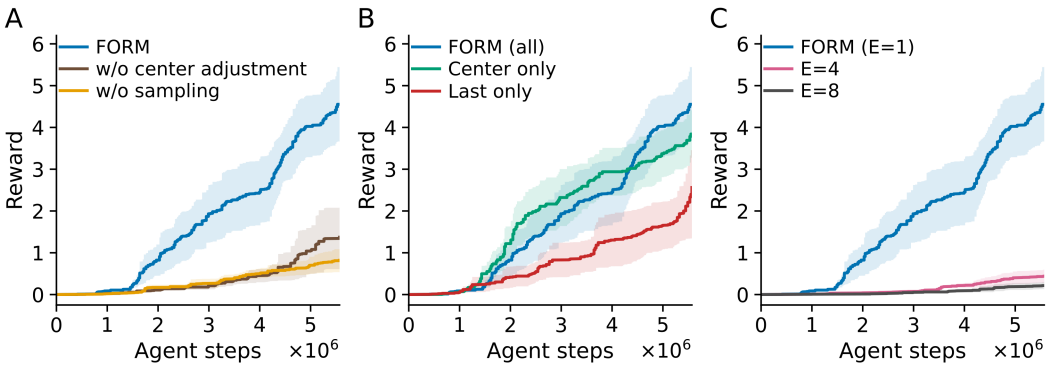

Figure 5: **Ablations of FORM design choices.** (**A**) *Morphology sampling and center allocation. w/o center adjustment* keeps 8 of 64 environments at the center morphology ($c = 0.125$), whereas *w/o sampling* assigns all environments to the center ($c = 1$). Both ablations reduce performance, with removing sampling causing the larger drop. (**B**) *Morphology-gradient sources. Center only* uses gradients only from the center cohort, whereas *Last only* uses only the final horizon of each episode. Including sampled robots provides a modest benefit, while aggregating gradients across the full episode provides a substantially larger improvement. (**C**) *Controller–morphology update timescale.* E denotes the number of controller-training episodes between consecutive morphology updates, with FORM using $E = 1$. Under the same interaction budget, holding morphology fixed for longer intervals ($E = 4$ or 8) substantially reduces performance. All curves show best-so-far reward across four seeds, with shading indicating standard error of the mean.

3.3 Ablations

We compare three groups of variants using four matched initialization seeds and a common budget of 5.57 million agent steps (Figure 5).

Morphology sampling and center allocation. FORM adaptively increases the center fraction as $c_k = 0.125 + 0.625\bar{q}_k$ during the mixed-population phase and sets $c_k = 1$ for the final 64 updates. We compare this schedule with a fixed $c = 0.125$ (*w/o center adjustment*) and with $c = 1$ throughout training (*w/o sampling*). These variants attain mean best-so-far returns of 1.37 and 0.82, respectively, compared with 4.55 for FORM (Figure 5(a)). Thus, maintaining neighboring sampled designs is important, while progressively reallocating environments toward the center provides an additional benefit.

Morphology-gradient aggregation. FORM aggregates morphology gradients across both the robot population and the N short horizons of each episode. Let $\mathcal{C}_k$ denote the center cohort at design-update episode k, with $|\mathcal{C}_k| = c_k B$. *Center only* replaces the population gradient with

$$g^{(n)}_{\mu,\text{center}} = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} g^{(n)}_{\mu,i},$$

while retaining sampled morphologies for controller training, and reaches 3.84 compared with 4.55 for FORM. *Last horizon only* instead uses $\bar{g}_\mu = g^{(N-1)}_\mu$ rather than averaging over all N horizons, and reaches 2.56 (Figure 5(b)). Gradients from sampled morphologies provide a modest improvement, whereas aggregating morphology gradients across all horizons has a substantially larger effect.

Controller–morphology update interval. Let E denote the number of controller-training episodes between morphology updates, with FORM using $E = 1$. For $E = 4$ or 8, each population remains fixed for multiple episodes and only the final episode contributes the morphology gradient. Under the same interaction budget, this reduces the number of morphology updates from 256 to 64 or 32, respectively. The resulting returns are 0.44 and 0.22 (Figure 5(c)), showing that longer morphology-hold intervals substantially reduce performance within the available training budget.

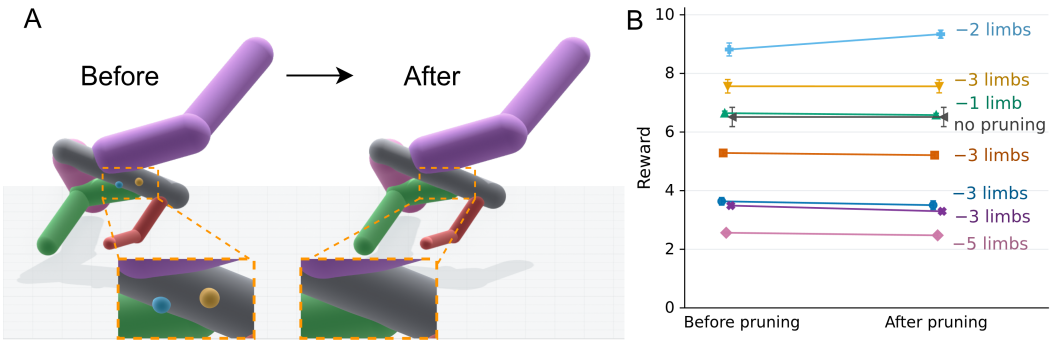

Figure 6: **Pruning vestigial limbs.** Very small limbs with presence $p_i < 0.1$ were removed to measure how (if at all) they contributed to behavior (**A**). The learned controller was not retrained after pruning; the observation and action channels of pruned limbs were masked to zero. Points show mean reward with standard-error bars of eight stochastic rollouts; annotations indicate the number of small limbs removed from each robot.

3.4 The effect of small limbs

We measured the contribution of very small appendages with $p_i < 0.1$ by pruning them from the evolved morphologies (Fig. 6). Pruning removes the two links, joints, and actuators of each selected appendage. We reuse the learned controller and its morphology-conditioning input, filling missing joint-state entries with zeros before normalization and discarding commands for removed joints.

Pruning removes 20 of the 48 candidate appendages across the eight designs, leaving between one and six appendages per robot. Seven designs lose one to five appendages, while one requires no pruning. We evaluate each original–pruned pair over eight paired stochastic rollouts using the same action-noise realizations. Across designs, mean episode return changes from 5.564 before pruning to 5.560 afterward.

We further assess these reduced designs with newly trained controllers (Fig. 7; Appendix D). After fixing each evolved morphology and training a fresh controller for 256 generations per environment, mean best-so-far returns reach 6.72 ± 0.93 (mean ± s.e.) before pruning and 6.96 ± 1.05 after pruning. Both exceed the Ant reference of 4.32 at the same agent-step budget.

4 Discussion

In this paper, we demonstrated for the first time the end-to-end differentiable co-design of morphology and control in agile rigid-bodied robots. Our approach (FORM; Fig. 2) efficiently generated a diversity of novel body plans (Fig. 4) and gaits (Fig. 1). Under matched interaction and wall-clock budgets, FORM yielded robots with much higher fitness than otherwise equivalent methods that approximated design gradients or assumed a quadrupedal form (Fig. 3).

There were several limitations of FORM which could be overcome in future work. First, the design space was restricted to capsule primitives and kinematic trees that could only branch from a single parent link. So a humanoid body plan with fingers branching from arms that branch from a torso could not evolve, as just one example. Robots with more intricate morphologies have been evolved (Li et al., 2025; Wang et al., 2026), but they did so inefficiently in non-differentiable environments.

A second, more important limitation was that the agile agents that evolved in this paper were not realized as physical robots (Yu et al., 2026). However, we did attempt to make the virtual environment and dynamics realistic, at least on visual inspection. Successful sim2real transfer is likely to require domain randomization (Peng et al., 2018) as well as system identification of joint dynamics and manual correction of mass, actuator torque, and joint range of motion, based on empirical measurements.

Despite these limitations, the results show that evolution can be represented as differentiable process in a much wider array of morphologies and behaviors than was previously believed to be possible.

REFERENCES

Alejandro M Castro, Ante Qu, Naveen Kuppuswamy, Alex Alspach, and Michael Sherman. A transition-aware method for the simulation of compliant contact with regularized friction. *IEEE Robotics and Automation Letters*, 5(2):1859–1866, 2020.

François Cochevelou, David Bonner, and Martin-Pierre Schmidt. Differentiable soft-robot generation. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 129–137. ACM, 2023.

Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11 (2):127–138, 2010.

Ignat Georgiev, Krishnan Srinivasan, Jie Xu, Eric Heiden, and Animesh Garg. Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2024.

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

Sehoon Ha, Stelian Coros, Alexander Alspach, Joohyung Kim, and Katsu Yamane. Joint optimization of robot design and motion parameters using the implicit function theorem. In *Robotics: Science and Systems (RSS)*, 2017.

Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 4455–4464. PMLR, 2020.

Ryo Kikuuwe, Naoyuki Takesue, Akihito Sano, Hiromi Mochiyama, and Hideo Fujimoto. Admittance and impedance representations of friction based on implicit euler integration. *IEEE Transactions on Robotics*, 22(6):1176–1188, 2006. doi: 10.1109/TRO.2006.886262.

Hiroki Kobayashi, Farzad Gholami, S Macrae Montgomery, Masato Tanaka, Liang Yue, Changyoung Yuhn, Yuki Sato, Atsushi Kawamoto, H Jerry Qi, and Tsuyoshi Nomura. Computational synthesis of locomotive soft robots by topology optimization. *Science Advances*, 10(30): eadn6129, 2024.

Muhan Li, Lingji Kong, and Sam Kriegman. Generating freeform endoskeletal robots. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Hod Lipson and Jordan B Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974, 2000.

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.

Pingchuan Ma, Tao Du, John Z Zhang, Kui Wu, Andrew Spielberg, Robert K Katzschmann, and Wojciech Matusik. Diffaqua: A differentiable computational design pipeline for soft underwater swimmers with shape interpolation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

Miles Macklin. Warp: A high-performance python framework for GPU simulation and graphics. https://github.com/NVIDIA/warp, March 2022. NVIDIA GPU Technology Conference (GTC).

David Matthews, Andrew Spielberg, Daniela Rus, Sam Kriegman, and Josh Bongard. Efficient automatic design of robots. *Proceedings of the National Academy of Sciences*, 120(41):e2305180120, 2023.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018.

Massimo Pigliucci. Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82, 2008.

Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to construct and control agents using deep reinforcement learning. In *2019 international conference on robotics and automation (ICRA)*, pp. 9798–9805. IEEE, 2019.

Andrew Spielberg, Brandon Araki, Cynthia Sung, Russ Tedrake, and Daniela Rus. Functional co-optimization of articulated robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pp. 5035–5042, 2017.

Luke Strgar, David Matthews, Tyler Hummer, and Sam Kriegman. Evolution and learning in differentiable robots. In *Robotics: Science and Systems (RSS)*, 2024.

Hyung Ju Terry Suh, Tao Pang, and Russ Tedrake. Bundled gradients through contact via randomized smoothing. *IEEE Robotics and Automation Letters*, 7(2):4000–4007, 2022.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

The Newton Contributors. Newton: GPU-accelerated physics simulation for robotics and simulation research. https://github.com/newton-physics/newton, 2025.

Yibin Wang, Muhan Li, Zihan Guo, and Sam Kriegman. Eco-moe: Embodiment-conditioned mixture of experts increases the evolvability of robots. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Richard A Watson and Eörs Szathmáry. How can evolution learn? *Trends in Ecology & Evolution*, 31(2):147–157, 2016.

Eliot Xing, Vernon Luk, and Jean Oh. Stabilizing reinforcement learning in differentiable multiphysics simulation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

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.

Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. In *International Conference on Learning Representations (ICLR)*, 2022.

Chen Yu, David Matthews, Jingxian Wang, Jing Gu, Douglas Blackiston, Michael Rubenstein, and Sam Kriegman. Agile legged locomotion in reconfigurable modular robots. *Proceedings of the National Academy of Sciences*, 123(10):e2519129123, 2026.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems (RSS)*, 2017.

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.

A ROBOT MORPHOLOGY

The candidate kinematic supergraph comprises a capsule-shaped root link and a set of candidate articulated appendages. Each appendage can be split into a proximal segment with hip hinge, and a distal segment with knee hinge. The root is encoded by the root dimensions (L_c, r_c). Each appendage i is represented using (L_i, r_i) for the nominal appendage size, (ζ_i, ϕ_i) for surface attachment, p_i for presence, $\beta_{m,i}, \beta_{c,i}$ for orientation, and η_i for hip axis.

Surface attachment chart. The root link is aligned with the x-axis and centered at the origin. We parameterize its surface by meridional arclength and circumferential angle. For appendage i, define

$$\xi_i = \zeta_i(L_c + \pi r_c), \qquad \xi_r = \frac{\pi r_c}{2}, \qquad \xi_f = \xi_r + L_c,$$

where ξ_i is measured from the rear pole and ξ_r, ξ_f mark the rear-cap/cylinder and cylinder/front-cap junctions of a capsule, respectively. The axial and radial coordinates are then

$$(x_i, \rho_i) = \begin{cases} \left(-\dfrac{L_c}{2} - r_c \cos\dfrac{\xi_i}{r_c},\ r_c \sin\dfrac{\xi_i}{r_c}\right), & \xi_i < \xi_r, \\ \left(-\dfrac{L_c}{2} + \xi_i - \xi_r,\ r_c\right), & \xi_r \le \xi_i \le \xi_f, \\ \left(\dfrac{L_c}{2} + r_c \sin\dfrac{\xi_i - \xi_f}{r_c},\ r_c \cos\dfrac{\xi_i - \xi_f}{r_c}\right), & \xi_i > \xi_f. \end{cases}$$

The corresponding surface attachment point is

$$\mathbf{u}_i = \begin{bmatrix} x_i \\ \rho_i \sin\phi_i \\ -\rho_i \cos\phi_i \end{bmatrix}$$

Let $\mathbf{t}_{m,i} = \partial \mathbf{u}_i / \partial \xi_i$ be the unit meridional tangent, and let $\mathbf{t}_{c,i} = (0, \cos\phi_i, \sin\phi_i)^\mathsf{T}$ be the unit circumferential tangent. The outward normal is $\mathbf{n}_i = \mathbf{t}_{c,i} \times \mathbf{t}_{m,i}$. Writing $\mathcal{N}(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|$, the deployed appendage direction $\mathbf{d}_i^{\text{dep}}$and hip axis $\mathbf{a}_{h,i}^{\text{dep}}$ are

$$\begin{aligned} \mathbf{d}_i^{\text{dep}} &= \mathcal{N}(\mathbf{n}_i + \tan\beta_{m,i}\,\mathbf{t}_{m,i} + \tan\beta_{c,i}\,\mathbf{t}_{c,i}), \\ \widetilde{\mathbf{a}}_{h,i} &= \cos\eta_i\,\mathbf{t}_{c,i} + \sin\eta_i\,\mathbf{t}_{m,i}, \\ \mathbf{a}_{h,i}^{\text{dep}} &= \mathcal{N}\Big(\widetilde{\mathbf{a}}_{h,i} - (\widetilde{\mathbf{a}}_{h,i}^\mathsf{T}\mathbf{d}_i^{\text{dep}})\mathbf{d}_i^{\text{dep}}\Big). \end{aligned}$$

These vectors define the right-handed deployed frame

$$R_i = \left[\mathbf{d}_i^{\text{dep}} \times \mathbf{a}_{h,i}^{\text{dep}} \quad \mathbf{a}_{h,i}^{\text{dep}} \quad -\mathbf{d}_i^{\text{dep}}\right],$$

whose columns are the local $+x, +y, +z$ axes. Thus the local $-z$ direction points along the appendage and the local $+y$ direction is the hip hinge axis.

Nominal two-segment geometry. The nominal appendage length is split into proximal (p) and distal (d) segments as

$$L_{p,i} = \alpha L_i, \qquad L_{d,i} = (1-\alpha)L_i, \qquad \alpha = 0.40.$$

For a capsule of cylindrical length L and radius r, its volume is $V(L, r) = \pi r^2 L + \frac{4}{3}\pi r^3$. We set the nominal segment radii to

$$\bar{r}_{p,i} = \lambda_i r_i, \qquad \bar{r}_{d,i} = \kappa\lambda_i r_i, \qquad \kappa = 0.85,$$

where the positive scale λ_i satisfies

$$V(L_{p,i}, \bar{r}_{p,i}) + V(L_{d,i}, \bar{r}_{d,i}) = V(L_i, r_i).$$

λ_i is computed using Newton's method.

Continuous presence and docking. Presence affects geometry and attachment through two distinct functions, presence gate $g(p)$ and docking weight $w(p)$, by defining

$$g(p_i) = g_{\min} + (1 - g_{\min})p_i, \qquad w(p_i) = \frac{p_i^n}{p_i^n + h^n},$$

with $g_{\min} = 0.25$, $h = 0.125$, and $n = 2$. The geometry gate scales both the length and radius of each segment:

$$\ell_{s,i} = g_i L_{s,i}, \qquad r_{s,i} = g_i \bar{r}_{s,i}, \qquad s \in \{p, d\}.$$

The docking weight controls the hip position, link orientation, and joint anchors. Define a docked frame

$$R_i^{\text{dock}} = [\mathbf{e}_x \times \mathbf{t}_{c,i} \quad \mathbf{t}_{c,i} \quad -\mathbf{e}_x],$$

where $\mathbf{e}_x = (1, 0, 0)^\top$ is the unit vector along the root $+x$-axis, so that its appendage direction is the root $+x$ axis. The actual proximal segment rest frame and hip pivot are

$$\widehat{R}_i = \text{Slerp}(R_i^{\text{dock}}, R_i; w_i), \qquad \mathbf{h}_i = w_i \mathbf{u}_i,$$

where interpolation is implemented using shortest-path spherical linear interpolation (SLERP). Thus, decreasing presence moves the hip toward the root center and rotates the appendage toward its docked orientation.

The continuous representation enables continuous optimization over both topological and geometric changes. Presence is decoded from a logit clamped to $[-3, 3]$, giving $p_i \in [\text{sigm}(-3), \text{sigm}(3)] \approx [0.0474, 0.9526]$. Here, sigm is the sigmoid function.

Rest-pose joint geometry. In the zero-joint-angle pose, the proximal and distal directions are

$$\widehat{\mathbf{d}}_{p,i} = -\widehat{R}_i \mathbf{e}_z, \qquad \widehat{\mathbf{d}}_{d,i} = -\widehat{R}_i \,\text{Rot}(\chi_i \mathbf{e}_x, b)\mathbf{e}_z,$$

where $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ are the standard basis vectors, $\chi_i \in \{-1, +1\}$ fixes knee chirality, and $\text{Rot}(\chi_i \mathbf{e}_x, b)$ applies the fixed rest bend $b = 55^\circ$ in the proximal frame. The hip and knee axes in the root frame are $\widehat{R}_i \mathbf{e}_y$ and $\chi_i \widehat{R}_i \mathbf{e}_x$, respectively. The anchor fractions along each cylindrical centerline satisfy

$$f_{hp,i} = f_{kd,i} = \frac{1 - w_i}{2}, \qquad f_{kp,i} = \frac{1 + w_i}{2},$$

with subscripts denoting the joint (h: hip, k: knee) and segment (p: proximal, d: distal). Thus the knee pivot is $\mathbf{k}_i = \mathbf{h}_i + w_i \ell_{p,i} \widehat{\mathbf{d}}_{p,i}$.

B ACTUATOR

We use position control with morphology-dependent motor capacities and PD gains. Each policy action specifies a target joint angle; the resulting PD torque is constrained by a symmetric torque–speed envelope and the appendage's presence. The hip and knee of appendage i share the same presence p_i.

Nominal motor capacity. Motors are sized using the nominal segment lengths $L_{p,i}, L_{d,i}$ and nominal masses $\bar{m}_{p,i}, \bar{m}_{d,i}$, before applying the geometry gate $g(p_i) = 0.25 + 0.75 p_i$. A fixed reference load is defined by

$$M_{\text{ref}} = \rho_c V(L_{c,\text{ref}}, r_{c,\text{ref}}), \qquad (L_{c,\text{ref}}, r_{c,\text{ref}}) = (0.2, 0.052)\,\text{m},$$

with root density ρ_c and the capsule volume V in Appendix A. This reference remains fixed as the actual root link dimensions change. This anchors motor sizing to a common load scale, preventing changes in root morphology from directly rescaling actuator capacity. With motor capacity coefficient $c_\tau = 4.0$ and gravitational acceleration $g_0 = 9.81\,\text{m}\,\text{s}^{-2}$, the nominal stall torques are

$$S_{h,i} = c_\tau g_0 \left[\frac{M_{\text{ref}}}{3}(L_{p,i} + L_{d,i}) + \frac{\bar{m}_{p,i} L_{p,i}}{2} + \bar{m}_{d,i} \left(L_{p,i} + \frac{L_{d,i}}{2} \right) \right],$$

$$S_{k,i} = c_\tau g_0 \left[\frac{M_{\text{ref}} L_{d,i}}{3} + \frac{\bar{m}_{d,i} L_{d,i}}{2} \right].$$

These load-based capacities depend on nominal appendage dimensions and are defined before presence scaling. For either joint j of appendage i, the effective joint armature is scaled with nominal motor capacity as

$$J_{a,j} = J_{\min} + (p_i)\gamma_J \frac{S_j}{\omega_0^2}, \qquad J_{\min} = 10^{-4}\,\mathrm{kg\,m^2}, \quad \gamma_J = 1.25, \quad \omega_0 = 20\,\mathrm{rad\,s^{-1}}.$$

The positive floor maintains nonzero armature throughout the design space.

Position targets and PD gains. The policy outputs a 12-dimensional action, with one component for each hip and knee joint. For joint j, let a_j denote its action component, q_j its angle, and $\omega_j = \dot{q}_j$ its angular velocity. The action is mapped to a target angle within the joint limits $[q_{\min,j}, q_{\max,j}]$ as

$$q_j^\star = q_{\min,j} + \frac{\mathrm{clip}(a_j, -1, 1) + 1}{2}(q_{\max,j} - q_{\min,j}).$$

We compute gains from conservative inertia proxies using the effective masses $m_{s,i}$ and lengths $\ell_{s,i}$ of (h: hip, k: knee) after geometry scaling:

$$\widehat{I}_{h,i} = (m_{p,i} + m_{d,i})(\ell_{p,i} + \ell_{d,i})^2 + J_{a,h,i}, \qquad \widehat{I}_{k,i} = m_{d,i}\ell_{d,i}^2 + J_{a,k,i}.$$

These are conservative inertia proxies rather than exact configuration-dependent joint inertias. The simulator also applies passive joint damping b_j. For position-loop natural frequency $f_p = 2$ Hz, angular frequency $\omega_p = 2\pi f_p$, and damping ratio $\zeta_p = 1$, the requested torque $\tau_{\mathrm{req},j}$ of joint j can be calculated by

$$K_{p,j} = \omega_p^2 \widehat{I}_j, \qquad K_{d,j} = \max(0, 2\zeta_p\omega_p\widehat{I}_j - b_j), \qquad \tau_{\mathrm{req},j} = K_{p,j}(q_j^\star - q_j) - K_{d,j}\omega_j.$$

Subtracting b_j accounts for damping already supplied by the simulator.

Torque–speed envelope and presence. With the speed-dependent capacity

$$H(\omega_j) = \max(0, 1 - |\omega_j|/\omega_0)$$

the active motor torque with presence p_i and nominal stall torque S_j is

$$\tau_{\mathrm{motor},j} = \mathrm{clip}(\tau_{\mathrm{req},j}, -p_i S_j H(\omega_j), p_i S_j H(\omega_j)).$$

The torque capacity is $p_i S_j$ at zero speed and vanishes when $|\omega_j| \geq \omega_0$.

Joint limits and differentiation. To maintain a comparable joint-limit response timescale across morphologies, we scale the numerical limit stiffness and damping with the joint inertia proxy:

$$k_{\mathrm{lim},j} = \omega_{\mathrm{lim}}^2\,\mathrm{sg}(\widehat{I}_j), \qquad b_{\mathrm{lim},j} = 2\zeta_{\mathrm{lim}}\omega_{\mathrm{lim}}\,\mathrm{sg}(\widehat{I}_j),$$

where $\omega_{\mathrm{lim}} = 150\,\mathrm{s^{-1}}$, $\zeta_{\mathrm{lim}} = 1$, and sg denotes stop-gradient. Since this morphology dependence serves only to scale the numerical joint-limit penalty, gradients through the coefficient scaling are removed to prevent morphology optimization from exploiting it. Other physical actuator and damping terms remain morphology-dependent and differentiable.

C SIMULATOR

We simulate articulated rigid-body dynamics with a Featherstone solver and compliant contact forces inside Newton simulator (The Newton Contributors, 2025). Each control interval of $1/60\,\mathrm{s}$ contains 32 physics substeps. Normal contact uses a penetration-dependent spring–damper model, while tangential friction uses a local implicit update followed by Coulomb saturation. For the flat-ground experiments, contacts are generated analytically between the robot's capsule endpoints and the ground plane.

Normal contact. Let $\delta = \max(-d, 0)$ be the penetration depth and v_n the relative normal velocity, positive for separation. Following DFlex (Xu et al., 2022),

$$F_n = k_e\delta + k_d\delta\max(-v_n, 0),$$

where the second term provides penetration-dependent damping only during approach.

Local tangential stabilization. For a contact with tangential relative velocity $\mathbf{u} \in \mathbb{R}^2$, let J_t be the point Jacobian expressed in the local tangent basis, and let $\mathcal{M}$ be the generalized mass matrix including joint armature. The tangential Delassus operator

$$W_t = J_t \mathcal{M}^{-1} J_t^\top \succeq 0$$

maps a tangential contact force $\boldsymbol{\phi}_t$ to the corresponding velocity change,

$$\Delta \mathbf{u} = \Delta t \, W_t \boldsymbol{\phi}_t.$$

Before Coulomb saturation, an explicit viscous friction law $\boldsymbol{\phi}_t^n = -k_f \mathbf{u}^n$ gives

$$\mathbf{u}^{n+1} = (I - \Delta t \, k_f W_t)\mathbf{u}^n.$$

Along an eigenvector of W_t with eigenvalue $\lambda \geq 0$, this update multiplies the velocity by

$$1 - \chi, \qquad \chi = \Delta t \, k_f \lambda.$$

Hence, when $\chi > 2$, the magnitude of the amplification factor exceeds one even though the corresponding continuous dynamics are dissipative. This can destabilize both the forward rollout and gradient propagation. The issue is particularly relevant during morphology optimization, where small or weakly expressed links can reduce effective contact mass and thereby increase λ.

To stabilize these local tangential dynamics, we instead use an implicit update. Following the use of implicit integration for frictional dynamics (Castro et al., 2020; Kikuuwe et al., 2006), we use the implicit update

$$\boldsymbol{\phi}_t^\star = -k_f \mathbf{u}^{n+1}, \qquad \mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \, W_t \boldsymbol{\phi}_t^\star.$$

Combining the two gives

$$(I + \Delta t \, k_f W_t)\mathbf{u}^{n+1} = \mathbf{u}^n.$$

For the same eigenvector, the amplification factor becomes

$$\frac{1}{1+\chi} \in (0, 1],$$

so the linear update cannot amplify tangential velocity.

The corresponding unsaturated trial force is

$$\boldsymbol{\phi}_t^\star = -k_f (I + \Delta t \, k_f W_t)^{-1} \mathbf{u}^n.$$

We then project $\boldsymbol{\phi}_t^\star$ onto the Coulomb disk $\|\boldsymbol{\phi}_t\| \leq \mu_f F_n$. The implicit update therefore requires only a 2×2 linear solve per contact, which is local to each contact.

Static preload initialization. Rather than performing a settling rollout, we initialize the robot with a small static contact preload. The lowest capsule endpoint is first placed at ground contact. We then compute a requested downward root translation δ_{req} satisfying

$$\sum_j k_j \max(\delta_{\text{req}} - h_j, 0) = m_{\text{tot}} g_0,$$

where h_j and k_j are the initial surface gap and effective normal stiffness of endpoint j and m_{tot} is the total robot mass. The applied translation is $\delta = \min(\delta_{\text{req}}, 5\,\text{mm})$. The preload solve is performed without gradient tracking, so the resulting root-height offset is treated as constant with respect to morphology. Morphology gradients during the subsequent rollout still propagate through the morphology-dependent geometry and joint transforms.

D FRESH-CONTROLLER RETRAINING OF LEARNED MORPHOLOGIES

To test whether FORM discovers morphologies that remain effective independently of their originally co-adapted controllers, we retrain a fresh controller from scratch on each learned morphology before and after pruning (Figure 7). Both variants learn to move substantially faster than the fixed Ant baseline and exhibit nearly identical learning curves, indicating that the learned bodies remain readily trainable with new controllers and that pruning does not noticeably influence their trainability.

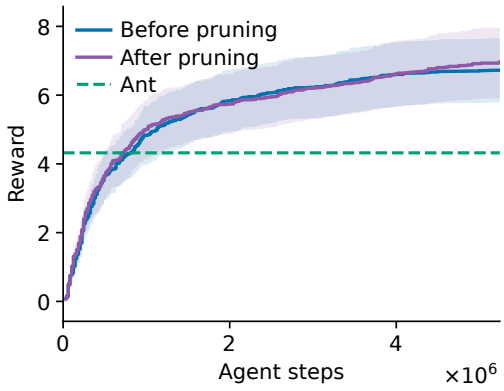

Figure 7: **Fresh-controller retraining of learned morphologies.** For each of the eight FORM runs, we fix the learned morphology before or after pruning and train a new controller from scratch for 256 episodes. Curves show the mean best-so-far reward across the eight learned morphologies, with standard-error shading. The dashed line denotes the best performance of the fixed Ant-inspired morphology under the same training budget. Before- and after-pruning morphologies exhibit nearly identical learning curves and both exceed the Ant baseline, showing that the learned bodies remain effective independently of their original co-adapted controllers and that pruning does not influence their capacity for learning.

E Additional parameters

Table 2 summarizes the shared controller and network settings. Table 3 gives the FORM morphology-optimization schedule. Tables 4 and 5 specify the simulation and morphology parameters.

Baseline and retraining schedules CMA-ES uses the same 16-episode warmup as FORM, followed by 378 generations to match FORM's wall-clock budget. Each generation contains 56 candidate environments and 8 center environments. The initial sampling standard deviations match the FORM sampling scales in Table 3; subsequent generations use the covariance adapted by CMA-ES. The fixed-morphology baselines are trained controller-only for 272 episodes. Fresh-controller retraining fixes the morphology and runs for 256 episodes without a separate warmup.

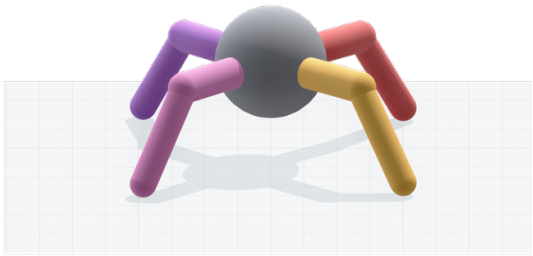

Figure 8: **Fixed Ant-inspired morphology used as a baseline.** Its morphology is held fixed during controller training; parameters are given in Table 5.

Table 2: Controller training and network settings

Parameter	Value
Rollouts and value fitting	
Parallel environments B	64
Episode length T / horizon H	320 / 8 control steps
Discount γ / TD parameter λ	0.9988454 / 0.95
Critic minibatch size / batches per pass	128 / 4
Fitting passes per horizon, each critic family	8
Controller optimization	
Actor / critics / temperature optimizer	AdamW, $(\beta_1, \beta_2) = (0.7, 0.95)$
Actor / critics / temperature learning rates	2×10^{-3} / 5×10^{-4} / 10^{-3}
AdamW weight decay	0.01
Actor learning-rate schedule / bounds	KL-adaptive / $[10^{-5}, 2 \times 10^{-3}]$
KL reference / adjustment factor	0.05 / 1.5
Actor / critics / temperature gradient-norm limit	0.5
Initial entropy temperature α_0	10^{-3}
Target entropy $\mathcal{H}_{\text{target}}$	$-\dim(\mathcal{A})/4$
Morphology target-critic EMA retention	0.95
Policy and value networks	
Actor / critic-member hidden widths	$(128, 64, 32)$ / $(64, 64)$
Ensemble members per critic family	2
Hidden-layer normalization / activation	LayerNorm / SiLU
Action distribution	Tanh-squashed Gaussian
Action log-standard-deviation bounds	$[-5, 2]$

Table 3: FORM morphology optimization and population schedule.

Parameter	Value
Controller warmup with fixed center	16 episodes
Design-update episodes	256: 192 mixed + 64 all-center
Policy episodes per morphology update E	1
Log std., non-presence coordinates	-1.5
Log std., presence logits	$\ln(0.5)$
Center allocation in mixed stages	Competence-based; fraction in $[0.125, 0.75]$
Competence reward reference / EMA retention	1.0 / 0.9
Design optimizer	Adam, $(\beta_1, \beta_2) = (0, 0)$
Morphology learning-rate schedule	Linear, $0.15 \to 0$
Morphology gradient-norm limit	5.0

Table 4: Simulation, contact, and actuation settings.

Parameter	Value
Integration and contact	
Control interval / physics substeps	$1/60\,\mathrm{s}$ / 32
Mass-matrix refresh interval	32 physics substeps
Effective Normal stiffness k_e,	$3 \times 10^4\,\mathrm{N\,m^{-1}}$
Normal damping parameter k_d	$500\,\mathrm{N\,s\,m^{-2}}$
Tangential coefficient k_f	$500\,\mathrm{N\,s\,m^{-1}}$
Friction coefficient μ_f	1.2
Contact candidate margin / shape margin	$0.02\,\mathrm{m}$ / $0.003\,\mathrm{m}$
Initial preload penetration bound	$0.005\,\mathrm{m}$
Actuation and physical properties	
PD frequency / damping ratio	2 Hz / 1
Motor capacity multiplier c_τ	4
No-load speed ω_0	$20\,\mathrm{rad\,s^{-1}}$
Minimum armature / multiplier	$10^{-4}\,\mathrm{kg\,m^2}$ / 1.25
Passive joint-damping rate	$5\,\mathrm{s^{-1}}$
Joint-limit frequency scale / damping ratio	$150\,\mathrm{s^{-1}}$ / 1
Hip angle limits	$[-\pi/4, \pi/4]\,\mathrm{rad}$
Relative-knee angle limits	$[-30^\circ, 50^\circ]$
Knee rest bend	55°
Fixed knee chirality $(\chi_1, \ldots, \chi_6)$	$(-1, +1, -1, +1, -1, +1)$
Root / appendage density	$(300, 250)\,\mathrm{kg\,m^{-3}}$
Proximal length / distal radius ratios	0.40 / 0.85

Table 5: Morphology initialization, design ranges, and fixed Ant parameters.

Parameter	FORM initial	FORM range	Ant (fixed)
Root length L_c (m)	0.14	$[0.10, 0.45]$	0
Root radius r_c (m)	0.04	$[0.035, 0.09]$	0.17
Appendage length L_i (m)	0.09	$[0.03, 0.75]$	$(0.557896, 0.18, 0.557896)$
Appendage radius r_i (m)	0.016	$[0.013, 0.075]$	0.0482436
Surface coordinate ζ_i	Randomized	$[0.02, 0.98]$	$(0.75, 0.50, 0.25)$
Attachment azimuth ϕ_i (rad)	Randomized	Periodic	$(\pm\pi/2, \pm 7\pi/30, \pm\pi/2)$
Presence p_i	0.12	$[0.0474, 0.9526]$	$(0.95, 0.12, 0.95)$
Meridional tilt $\beta_{m,i}$ (rad)	Randomized	$[-\pi/4, \pi/4]$	0
Circumferential tilt $\beta_{c,i}$ (rad)	Randomized	$[-\pi/4, \pi/4]$	0
Hip-axis orientation η_i (rad)	Randomized	$[-\pi/3, \pi/3]$	0

Triples denote front, middle, and rear appendage pairs; $\pm$ indicates left/right symmetry. Ant has a spherical root by setting ($L_c = 0$). The presence range is rounded from $[\mathrm{sigm}(-3), \mathrm{sigm}(3)]$.

GENERATIVE AI USAGE

We used generative AI tools to assist with method implementation, editing this paper for readability, and development of plotting and visualization scripts. For all other tasks requiring disclosure under the AI Policy for Authors, generative AI was not used. We reviewed all AI-assisted contributions for correctness and take responsibility for the final contents of this paper.