2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Co-Design and Morphology-Guided Feedback Control: An Approach for Soft Robots

Nhan Huu Nguyen[1], Truong Dinh Do[1], Minh Le Nguyen[1], and Van Anh Ho[1]

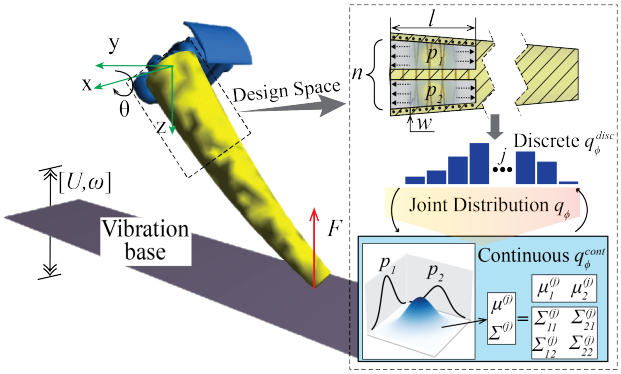
***Abstract*— Soft robots, with their highly compliant bodies, exhibit numerous unforeseen configurations that often defy engineering intuition and complicate control design. This work introduces a simulation-based co-optimization framework that jointly optimizes both morphology and control. Unlike existing approaches that rely on oversimplified soft robot models or feed-forward controllers for simple tasks, our framework targets complex tasks that benefit from closed-loop feedback. The controller is trained over a hybrid design space combining discrete parameters, which define the nominal structure, and continuous parameters, which shift the morphology adaptively. The design distribution is iteratively manipulated to emphasize high-performing candidates until the optimal design–control pair emerges. Proprioceptive feedback in the form of mechanical strain is integrated to provide the controller with awareness of morphological state and interaction dynamics. Demonstrations show that the framework converges reliably to optimal design–control solutions, validating the effectiveness of the proposed joint optimization strategy.**

I. INTRODUCTION

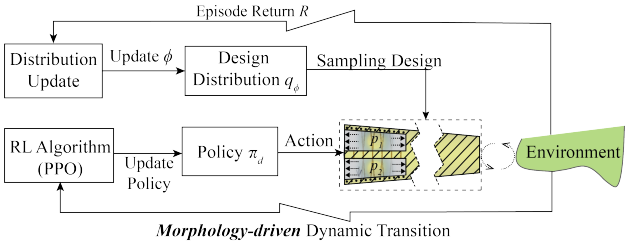
Taking inspiration from biological organisms, many successful robotic solutions seem to rely on a close and concurrent coupling between morphology and control. In practice, it is not just the body and the controller, but also the flow of perceptive information, that need to work together [1]. This interplay may be straightforward in rigid systems, but in soft robots it becomes far more intricate, and arguably more important. For this reason, reliable co-design frameworks tailored to soft bodies remain in high demand. Although joint design–control optimization has been studied extensively for rigid robots, progress in the soft robotics community is still limited. The result is that current soft robots fall short of what might be called *morphological intelligence*, a level of embodied adaptability that natural organisms display almost effortlessly.

Several issues remain unresolved, but in this study, we focus on three bottlenecks:

1) **Enormous and multi-dimensional design space**: Unlike rigid robots, which can be reduced to a finite set of joints and links, soft bodies are often described as a continuum of deformable elements. This description may be physically accurate, yet it makes systematic exploration extremely costly. To make progress, the design space is usually scaled down, often expressed in terms of a limited set of *discrete* or *continuous* parameters.
2) **Strong covariance among morphological variables**: Even after simplifying the design space, morphological variables in soft robots remain tightly interdependent.
3) **Lack of closed-loop feedback integration**: Most current studies still rely on feed-forward control, leaving out motor-sensory loops. On one hand, this supports the appealing notion of mechanical intelligence, where behavior arises from the natural body–environment interaction. But in reality, such an open-loop strategy is effective only for relatively simple tasks such as periodic walking or hopping. When conditions vary, morphology that can actively adapt through mechanisms like adjustable stiffness [2], [3] or pressure modulation [4] has been shown to improve robustness and even enable new behaviors. Bringing this adaptability into a closed-loop framework, through proprioceptive or exteroceptive feedback, may allow the system to respond dynamically to its own changing state, guiding optimization toward regions of the design space that



(a) Investigation scenario and design space modeling.



(b) Co-optimization framework driven by morphological-based feedback.

Fig. 1: Overview of the proposed framework for jointly design-control co-optimization.

This work was fully supported by JSPS Grant-in-Aid for Scientific Research (KAKENHI) Projects No.23K19096 and No.25K17569.

[1]Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan.
Corresponding author: Nhan Huu Nguyen. Email: nhnhan@jaist.ac.jp

are both feasible and resilient.

Motivated by these challenges, we present a simulation-based co-design framework developed specifically for soft robots whose bodies are described by a mix of discrete and continuous parameters. The framework relies on a hierarchical loop architecture: an outer loop that proposes and evaluates candidate morphologies, gradually concentrating on the more promising ones, and an inner loop that applies deep reinforcement learning (RL) to train control policies conditioned on these morphologies. As training progresses, evidence from policy performance returns to the outer loop, which in turn reshapes the sampling distribution. In this way, design evaluation guides policy learning, while control outcomes refine design choices. Over time, the search may converge toward regions of the morphology–control space that are both feasible and high-performing, rather than scattering effort across less relevant areas.

The distinctive feature of this work lies in how the design space is constructed. It accommodates both discrete descriptors of structure and continuous actuation variables that can reconfigure the nominal body shape defined by those discrete descriptors. In practice, these continuous parameters allow the robot's morphology to adapt online, yielding what may be called an *adaptive morphology*. The resulting changes in body configuration are sensed and reintroduced into the control loop through proprioceptive feedback. We suggest that this additional layer of adaptation accelerate convergence during optimization and, more importantly, make the framework extendable to more demanding tasks. Beyond optimization, the framework also serves as a testbed: a controlled environment where algorithmic strategies and engineered solutions can be evaluated, and unexpected structures or behaviors may emerge from the interplay of design and control.

II. Related Works

Given the importance of co-optimization in robot development, a large body of work has focused on rigid robots [5]–[11]. Most approaches adopt an inner–outer loop structure, where evolutionary methods or gradient-based optimization are used to refine designs [9], [11]–[14]. These methods are attractive because they can explore design spaces thoroughly and escape local optima. The trade-off, however, is that convergence often requires long training cycles. More recently, reinforcement learning has been explored as a way to co-develop robot structures and controllers simultaneously [7], [8], [15]. Another direction frames co-design as a graph search problem, which allows graph-based algorithms to be applied to design optimization [15]–[18].

By contrast, work on co-optimization for soft robots is still at an early stage, though notable progress has been made. For rigid robots, the dimension of the design space is typically bounded by the number of rigid parts. In soft robots, the compliant nature of the body often increases the effective design space manyfold. A common strategy in preliminary studies is to discretize the soft body into voxels and perform optimization in a simulation environment [19]–[21]. For example, Spielberg *et al.* used a differentiable simulator to reduce the dimensionality of voxel-based designs before optimization. In [20], sensor placement was treated as the optimization objective, under the assumption that proper distribution of discrete sensors across the body would yield a better representation of body dynamics, and thus better performance. Similarly, Bongard *et al.* [22] explored voxel deformation as a way for robots to compensate for physical damage, though their controller remained fixed. Recognizing the lack of standard benchmarks, Bhatia *et al.* [23] introduced *Evolution Gym*, the first large-scale platform for soft robot co-optimization. Building on this, Wang *et al.* [24] proposed a co-design framework tailored for locomotion in diverse terrains. While these contributions represent important milestones, they almost exclusively focus on feed-forward control schemes and simple behaviors, such as walking. Furthermore, voxel-based representations differ substantially from most soft robotic devices described in the literature, raising questions about their transfer to actual applications.

More recent efforts have moved toward designs that better match real soft actuators. Schaff *et al.* [25], [26] presented a co-design framework and a sim-to-real transfer pipeline for a soft crawler with PneuNet-based legs. In this case, optimization focused on the layout of the PneuNet chambers rather than the intrinsic geometry of each leg. Soft robot morphology, however, need not be static. It can be defined by a combination of discrete and continuous morphology-change activation parameters. Inspired by many works on adaptive morphology, for instance [2], [22], [27], we adopt a paradigm in which the nominal geometry is set by discrete parameters, but can be shifted online through continuous actuation. This transformation does more than alter the robot's appearance, but it changes mechanical characteristics such as stiffness, which in turn are sensed and fed back to the controller. Such a design philosophy reframes the soft body as an actively transformable medium. Rather than being a fixed constraint, morphology becomes a resource for adaptation, potentially giving rise to robust and unexpected behaviors that go beyond traditional engineering intuition.

III. Problem Description

A. Investigation Scenario

The proof-of-concept setup for our framework is illustrated in Fig. 1a. A soft beam, implemented with our whisker-like tactile sensor [2], [27], [28], is fixed at its proximal end to a rotary base. The base sets the rotation angle, denoted by θ. The distal end of the beam makes contact with a horizontally flat vibrating plate characterized by oscillation amplitude U and pulsation ω, serving as an analogue to a robot leg interacting with the ground. The beam is initially positioned at an inclination angle α relative to the plate. When the plate vibrates, the beam undergoes contact reactions that can bend the body. We denote this contact force as $\mathbf{F}$ and refer specifically to its vertical component, F, since it dominates the interaction.

The control objective is to regulate F toward a prescribed target F_t by adjusting θ through a feedback policy. Performance is evaluated by measuring how well F tracks F_t. A tolerance band of 20% is enforced, and deviations outside this range incur an additional penalty (the exact weighting is specified later). In short, the aim is to develop a policy for θ that keeps $|F - F_t|$ as small as possible while maintaining stability within the tolerance band.

The soft beam itself is constructed as a cone-shaped elastomeric bar. Its larger end houses one or more pneumatic chambers that can be pressurized to locally alter morphology and compliance. In this study, we focus on axial strain ϵ as the proprioceptive signal driving feedback control. To ensure axial strain dominates, inextensible fibers are helically wound around the chamber region. This reinforcement permits axial elongation while constraining radial expansion. The resulting strain measurements are fed back to the controller regulating θ, thereby closing the loop and capturing the morphology-dependent dynamics of the system.

B. Geometry Parameterization

The soft whisker's geometry is parameterized by two categories of variables: (i) discrete structural variables, which define the nominal morphology of the chamber region; and (ii) continuous actuation variables, which modulate this morphology during operation. Following [27], we consider three discrete parameters: the number of chambers $n \in 1, 2$; the chamber length $l \in [20, 40]$ mm in increments of 2 mm; and the wall thickness $w \in [2, 4]$ mm in increments of 0.5 mm. Taken together, these choices yield a finite design set of 110 unique configurations, indexed as $j \in \mathcal{J} = [1 \dots 110]$. Additional details of the whisker's geometry are provided in [27].

For each discrete design $j = n, l, w$, the local chamber morphology can be actuated by internal pressures $p = p_1, p_2$, each bounded within $[0, 20]$ kPa. Independent control of p_1 and p_2 is possible when $n = 2$, while in the case of $n = 1$, p_2 is set to zero. The full design vector is therefore $d = j, p$, where $d \in \mathcal{D}$ defines the overall design space. Adjusting p reshapes compliance and geometry, resulting in an adaptive morphology relative to the nominal configuration j. Alternative parameterizations are also possible. For instance, l and w could be treated as continuous variables within manufacturing tolerances [27], or conversely, p could be discretized to reduce computational load. Such flexibility does not change the generality of the framework but allows the formulation to adapt to different experimental or computational constraints.

IV. Methodology

A. MDP Formulation and Learning Environment Setup

The co-optimization problem can be formulated as a Markov decision process (MDP) which is mathematically represented by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition function $\mathcal{P} : s \in \mathcal{S} \times a \in \mathcal{A} \times s' \in \mathcal{S} \to [0, 1]$, where s and s' are the current state and the next state, respectively, $\mathcal{R}$ is the reward function.

We define $s = \{j, p, \epsilon\}$. By treating morphology $\{j, p\}$ as an observed variable, the policy π_θ adapts its control law to the body's instantaneous configuration and its coupled dynamics (reflected by mechanical strain ϵ), rather than relying solely on long-horizon reward signals. Each state variable is normalized to the range $[-1, 1]$ using min–max scaling, with reference values determined in simulation. The action a is a continuous command subjected to the range $[-1, 1]$, representing a normalized rotation command. The actual rotation for the next step is computed as: $\Delta\theta = a \cdot \theta_{max}$, where $\theta_{max} = \frac{\pi}{18}$ rad denotes the maximum allowable rotation angle in this work.

The reward R is designed to encourage the agent to maintain the contact force F close to the target F_t. It is composed of three terms: a force-tracking reward r_f, a smoothness penalty r_{smooth}, and a flip-direction penalty r_{flip}. The first term is given by a raised cosine function when F is within the tolerance band:

$$\begin{aligned} &r_f = [0.5 \cdot (1 + \cos(\pi\delta))]^\kappa, \\ &\text{where } \delta = \tfrac{|F - F_t|}{0.2F_t}. \end{aligned} \tag{1}$$

Otherwise, $r_f = 0$. Equation 1 acts as a smooth, symmetric shaping function centered at F_t. The reward attains its maximum when $F = F_t$ and decreases gradually toward zero as F approaches the tolerance boundaries. The coefficient κ controls the steepness of this decay. In this work, we chose $\kappa = 1.5$ to produce a sharper peak at F_t to encourage precise control. First penalty element is calculated as follows:

$$r_{smooth} = w_{smooth} \cdot \frac{|\Delta\theta|}{\theta_{max}}, \tag{2}$$

where $w_{smooth} = 0.05$. This function provides the reduction in reward due to large rotation increments $\Delta\theta$ relative to the maximum allowable increment θ_{max}. Another penalty term is applied when the rotation direction changes sign between consecutive steps.

$$r_{flip} = w_{flip} \cdot \mathbf{1}\left[\text{sign}(\Delta\theta_t) \neq \text{sign}(\Delta\theta_{t-1})\right], \tag{3}$$

where $w_{flip} = 0.1$. Additionally, two termination conditions are enforced: (i) loss of ground contact, and (ii) excessive bending leading to buckling, both triggered by extreme θ values. The final form of reward calculation function is: $R = r_f - r_{smooth} - r_{flip}$.

B. Co-optimization Framework

Instead of training a separate controller for every design, we are inspired by the idea in [25] that treats co-optimization as learning a design-conditioned policy $\pi_\theta(a|s, d)$, *i.e.,* $\pi_d(\cdot \mid s) := \pi_\theta(\cdot \mid s, d)$ for brevity. Alongside this policy, we learn a distribution over designs $q_\phi(d)$. Each design $d \in \mathcal{D}$ induces a specific MDP, $\mathcal{M}_d = (\mathcal{S}_d, \mathcal{A}_d, \mathcal{P}_d, \mathcal{R})$. Our objective is to maximize expected return over the design distribution while encouraging exploration with an entropy regularizer. The optimal pair of controller and design, $\{\pi^*_{d^*}, d^*\}$, is obtained by solving:

$$d^*, \pi^*_{d^*} = \max_{\theta,\phi} \mathbb{E}_{d\sim q_\phi}\left[J(\pi_\theta; \mathcal{M}_d)\right] + \tau\mathcal{H}(q_\phi), \tag{4}$$

Algorithm 1 Co-optimization of Design and Control

Input: Design space $\mathcal{D}$, design distribution q_ϕ horizon T, batch size ψ, PPO & sim hparams
Output: $d^* = \{j^*, p^*\}$ and policy $\pi^*_{d^*}$

```
Init q_φ(β_0, R̂_0, μ_0, Σ_0); t ← 0, k ← 0
while t < T do
    Sample {d_{k,i} = {j_{k,i}; p_{k,i}}}_{i=1}^{ψ} with j_{k,i} ~ q_φ^disc,
    p_{k,i} ~ q_φ^cont(· | j_{k,i})
    Deploy for t_deploy, collect transitions; update θ with PPO
    Collect R_{d_k} = {R_{d_{k,i}}}; update q_φ via Alg. 2
    t ← t + ψ t_deploy; k ← k + 1
end while
```

where $J = \mathbb{E}_{\pi_d}\left[\sum_t \gamma^t R_t\right]$ is objective function, $\tau \geq 0$ controls the exploration capability of the design space, and $\mathcal{H}(q_\phi)$ refers to the entropy of the design distribution space.

Intuitively, q_ϕ starts out broad, sampling across diverse morphologies. Over the simulation time, it is annealed, gradually concentrating probability mass on higher-return designs, while the policy adapts to specialize in those regions. In practice, at each episode ψ design samples are drawn and evaluated in parallel for data collection. Once ψ designs have been explored, q_ϕ is updated to reflect their outcomes. More specifically, the training process for T number of steps will be divided into three phases:

1) *Exploration* ($0 \leq t < T_1$). τ is set to encourage a nearly uniform q_ϕ, so designs are drawn from a wide pool.
2) *Transition* ($T_1 \leq t < T_2$). q_ϕ is gradually reweighted toward designs with higher returns under the current policy. At the same time, π_θ is updated on increasingly promising subsets of designs.
3) *Exploitation* ($T_2 \leq t \leq T$). The distribution collapses to a top-performing design d^*, and the controller is fine-tuned exclusively on $\mathcal{M}_{d^*}$ to produce $\pi^*_{d^*}$.

This hierarchical schedule allows q_ϕ guiding sampling toward high-performing morphologies while π_θ acquires competence that transfers across related designs, ultimately yielding a specialized controller for the selected design. Algorithm 1 summarizes the workflow of the proposed method.

C. Distribution Modeling and Update Strategy

This section details how the design distribution q_ϕ is modeled and updated based on performance across generations. Because the design space $\mathcal{D}$ is heterogeneous—containing both discrete and continuous variables—the main challenge is to capture correlations among them without making the model intractable. To simplify, we factorize q_ϕ into two components: q_ϕ^{disc} and q_ϕ^{cont}, corresponding to discrete and continuous variables, respectively. Updates are synchronized with the evaluation of fixed-size batches of sampled designs. Each component is updated independently. We define k as the number of distribution updates performed during training.

To achieve this heterogeneous nature, our strategy is first to create a subspace for discrete design parameters G as a categorical (Gibbs) distribution q_ϕ^{disc}. As suggested in [25], it is efficient to control the pressure distribution across the categories by setting the logits $z = \beta R_d$ into the Softmax function.

$$q_\phi(j) = \frac{e^{\beta \widehat{R}^{(j)}}}{\sum_{j \in \mathcal{J}} e^{\beta \widehat{R}^{(j)}}}, \tag{5}$$

where β is the inverse temperature and $\widehat{R}$ is reward-driven score reflecting the performance of design j. Manipulating β controls exploration: $\beta = 0$ yields a nearly uniform distribution (exploration), while large β sharpens the focus on high-return designs (exploitation) as introduced in Section IV-B. To update this distribution model, we substitute $\widehat{R}$ in Eq. 5 with $\widehat{R} = [\widehat{R}^{(j)}]$, as the set of categorical reward scalars corresponding to the discrete design space. In the *Exploration* stage, $\widehat{R}^{(j)}$ is defined as the best accumulated reward observed for design j. During *Transition*, it becomes a moving average (computed with RMS) over a rolling buffer of recent returns R_d. This history-aware approach prevents the algorithm from discarding potentially strong designs due to short-term noise.

Next, a multivariate normal distribution for the continuous correlated pressures p, conditioned by the discrete choice, $q_\phi^{cont}(p \mid j)$ is built up. We modeled $q_\phi^{cont}(p \mid j) = \mathcal{N}\left(p; \mu^{(j)}, \Sigma^{(j)}\right)$ which is characterized by mean variables $\mu^{(j)} = [\mu_1^{(j)}, \mu_2^{(j)}]$ and corresponding covariances

$$\Sigma^{(j)} = \begin{bmatrix} \Sigma_{11}^{(j)} & \Sigma_{21}^{(j)} \\ \Sigma_{12}^{(j)} & \Sigma_{22}^{(j)} \end{bmatrix} = \mathcal{L}^{(j)} \mathcal{L}^{(j)T}, \tag{6}$$

where $\mathcal{L}^{(j)}$ is the Cholesky factor ensuring positive definiteness. An issue arises when the number of chambers is $n = 1$. Setting $p_2 = 0$ is straightforward, but the off-diagonal covariance terms $\Sigma_{12}^{(j)}$ and $\Sigma_{21}^{(j)}$ may still couple p_1 and p_2. To properly decouple them, we explicitly zero out these off-diagonal entries whenever $n = 1$. Initially, the continuous distribution is initialized to have random means and high variance, *i.e.,* high entropy. During Exploration, the design distribution remains uniform while the policy is updated. As training progresses toward Exploitation, q_ϕ^{cont} is updated based on the observed rewards for each design $d = (j, p)$. In details, each sampled design $d = [j, p]$ with the earned reward R_d, the expected mean and the covariance matrix at the $k+1$ update are computed by:

$$\begin{aligned} \mu_{k+1}^{(j)} &= \mu_k^{(j)} + \left(R_k^{(j)} - b_k^{(j)}\right) \nabla_{\mu_k^{(j)}} \log q_\phi^{\text{cont}}(j) \\ &= \mu_k^{(j)} + \alpha \left(R_k^{(j)} - b_k^{(j)}\right) \left(p(j) - \mu_k^{(j)}\right), \end{aligned} \tag{7}$$

$$\begin{aligned} \Sigma_{k+1}^{(j)} &= \Sigma_k^{(j)} + \left(R_k^{(j)} - b_k^{(j)}\right) \nabla_{\Sigma_k^{(j)}} \log q_\phi^{\text{cont}}(j) \\ &= \Sigma_k^{(j)} + \alpha \left(R_k^{(j)} - b_k^{(j)}\right) \Big[\left(p(j) - \mu_k^{(j)}\right)\left(p(j) - \mu_k^{(j)}\right)^T \\ &\quad - \Sigma_k^{(j)}\Big], \end{aligned} \tag{8}$$

where $b_k^{(j)}$ is a baseline return for variance reduction. In this sense, the term $R_k^{(j)} - b_k^{(j)}$ thus acts like an advantage estimate, emphasizing updates for designs performing above their expected baseline while down-weighting weaker ones. After being used for Eq. 7 and 8, $b^{(j)}$ will be updated for the next update $k+1$ as follows:

$$b_{k+1}^{(j)} = (1-\varphi)\, b_k^{(j)} + \varphi R_k^{(j)}\,, \tag{9}$$

where φ controls how quickly the baseline tracks new rewards. Besides, α is a constant coefficient defining how strong one sample d with the advantage $R_k^{(j)} - b_k^{(j)}$ affect the mean $\mu_{k+1}^{(j)}$ and the covariance $\Sigma_{k+1}^{(j)}$. In our implementation, we set $\alpha = 0.1$ and $\varphi = 0.05$.

Algorithm 2 Update Method for Design Distribution

Input: d_k=$\{d_{k,i}\}_{i=1}^{\psi}$, R_{d_k}, α, φ, T_1, T_2; state $\beta_k, \mu_k, \Sigma_k, t, b_k, \widehat{R}_k$, RMS
Output: β_{k+1}, $\widehat{R}_{k+1}$, μ_{k+1} and Σ_{k+1}

for $d_{k,i} \in d_k$ **do**
  $j \leftarrow \text{index}(d_{k,i})$, $p \in \mathbb{R}^2$, $R \leftarrow R_{d_{k,i}}$
  UPDATERMS(j, r_{buffer}, R) ▷ update running stats via reward buffer
  if $t < T_1$ **then**
    if $R > \widehat{R}_k^{(j)}$ **then**
      $\widehat{R}_k^{(j)} \leftarrow R$ ▷ best-so-far
      $\mu_k^{(j)} \leftarrow \mu_0$, $\Sigma_k^{(j)} \leftarrow \Sigma_0$ ▷ no update
    end if
  else if $T_1 \le t < T_2$ **then**
    $\widehat{R}_k^{(j)} \leftarrow \text{RMS}^{(j)}.\text{mean}$ ▷ assign running mean
    $A \leftarrow \dfrac{R - b_k^{(j)}}{\text{RMS}^{(j)}.\text{std}}$ ▷ normalize advantage
    $\mu_{k+1}^{(j)} \leftarrow \mu_k^{(j)} + \alpha A\,(p - \mu_k^{(j)})$ ▷ Equation 7
    $\Sigma_{k+1}^{(j)} \leftarrow \Sigma_k^{(j)} + \alpha\, A\Big[(p-\mu_k^{(j)})(p-\mu_k^{(j)})^T - \Sigma_k^{(j)}\Big]$ ▷ Equation 8
    if $p_2(j_i)$=0 **then** zero $\Sigma_{12}^{(j)}$ and $\Sigma_{21}^{(j)}$ **end if** ▷ Single-chamber case
  end if
  $q_\phi(j \mid t) \leftarrow \text{softmax}_j\big(\beta(t)\,\widehat{R}_k^{(j)}\big)$
  $b_{k+1}^{(j)} \leftarrow (1-\varphi)\, b_k^{(j)} + \varphi R$
end for

V. Experiment and Results

A. Simulation and RL Environment

This section briefly introduces the simulation-based reinforcement learning environment in which all samples are modeled and deployed. The dynamics model of the soft leg is constructed in *SOFA*[1] - a multi-physics engine based on Finite Element Method (FEM). We adopted a linear constitutive model for the elastic material, which is governed by Young's modulus (E) and Poisson's ratio ν). To deal with the large deformation state of the soft body, a co-rotational FEM formulation is applied [29]. The generic dynamic equation is shown below:

$$\mathbf{M}\ddot{\mathbf{x}} = -dt^2\mathbf{K}(\mathbf{x_{t-1}})\dot{\mathbf{x}} + \mathbb{P} + \mathbb{F}(\mathbf{x_{t-1}}) + \mathbf{dt}\mathbf{J}^{\mathbf{T}}\boldsymbol{\lambda}\,, \tag{10}$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{x}$ elements nodes, $\mathbf{K} = \frac{\partial(\mathbf{x}_{t-1})}{\partial \mathbf{x}}$ is the stiffness matrix, $\mathbb{P}$ is known external forces (*e.g.,* gravity), $\mathbb{F}$ is internal force and $\mathbf{J}^T\boldsymbol{\lambda}$ is a vector that gathers constraint forces such as contact force, pressure or motor actuation forces.

To capture the effect of inextensible fibers, we follow the method introduced in [30], where each fiber is discretized into a chain of springs with specified stiffness. The number of springs is chosen to balance two objectives: providing sufficient constraint on radial expansion and keeping computational cost manageable. Both the nominal chamber geometry and the embedded fibers are meshed in GMSH before being imported into the SOFA simulation environment.

For reinforcement learning, we integrate the simulation with the SOFAGym package [31], which connects SOFA to the OpenAI Gym interface. In this study, we employ the Proximal Policy Optimization (PPO) algorithm, though our framework is not restricted to this specific choice. Each training episode runs for 250 simulation steps. A single control action is held constant over 5 steps, so each episode produces 50 reward evaluations. The controller π_d is parameterized by a multilayer perceptron (MLP). The main PPO hyperparameters and SOFA simulation settings are summarized in Table I and Table II. Material parameters such as Young's modulus and Poisson's ratio are tuned based on estimates from [29], ensuring both realism and numerical stability. To highlight the algorithmic aspects of our method, we assume frictionless contact at the leg tip, which simplifies interaction dynamics during training.

Hyperparameter	Value
Total timesteps	1 million
Learning rate	0.003
Number of steps	50
Batch size	256
Number of epochs	10
Discount factor (γ)	0.99
Clipping range	0.2
Entropy coefficient	0.01
Value function coefficient	0.6
Max gradient norm	0.5
Target KL	0.02
Number of sampled designs required to trigger updating	40
The batch size of sampled designs used for update	40
Entropy linear decay start	220K
Entropy linear decay end	800K
Policy network (π)	$[512, 512, 512]$
Optimizer	Adam

TABLE I: PPO Hyper-parameters for Co-optimization Algorithm

B. Results

Figure 2a summarizes the training progress of the proposed co-optimization framework. The histogram shows the

[1]Simulation Open Framework Architecture: www.sofa-framework.org

Hyperparameter	Value
Gravity	9800 mN
Young's modulus	500 kPa
Poisson ratio	0.4
Friction coefficient	Frictionless
Body mass	20 g

TABLE II: SOFA Modeling Parameters

reward distribution of designs sampled from q_ϕ, overlaid with a moving-average reward curve (red, window size 100) and a top-score curve (blue), which tracks the best reward achieved up to each time step t. Table III reports the optimal design d^* identified in the *Exploitation* phase, along with a comparable competitor d^{comp} (the second-most dominant design). These designs correspond to discrete indices $j = 21$ and $j = 63$, denoted as j_{21} and j_{63}, respectively. Figure 2b further illustrates the evolution of the mean μ_1 and standard deviation σ_1 of the continuous parameter for d^*, mapped back from the latent covariance domain to the physical actuation space. Since d^* is a single-chamber design, only μ_1 and σ_1 are displayed.

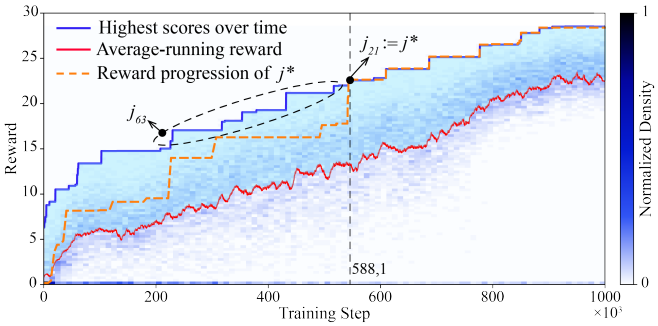
At the *Exploration* stage ($0 \leq t \leq$ 220K steps), the design distribution exported candidates nearly uniformly. Despite almost half of these designs yielding zero reward, overall performance improved steadily. This is likely due to structural similarities among candidates, which enabled the controller to generalize across them. improvements continued but became increasingly concentrated on high-performing designs. Importantly, optimization did not stall at a local optimum such as j_{63}. Although q_ϕ^{disc} placed strong sampling pressure on j_{63} during the middle of training, j_{21} consistently gained momentum and eventually overtook it at step 558.1K, as shown by the dashed yellow trajectory in Fig. 2a. This shift is further supported by Fig. 3a, which visualizes the reward distributions: j_{21} exhibited tighter, more consistent performance, while j_{63} remained broader and less stable. Meanwhile, the continuous parameters of d^* converged quickly (Fig. 2b), demonstrating both the efficiency of Algorithm 2 and the robustness of the hybrid distribution update.

Design	Discrete values [mm]	Continuous values [kPa]
d^*	$j^* \equiv j_{21} = \{1, 28\,2.5\,\text{mm}\}$	$p^* = \{0.069, 0\}$
d^{comp}	$j_{63} = \{2, 22\,\text{mm}, 3.5\,\text{mm}\}$	$p_{63} = \{0.0078, 0.0125\}$

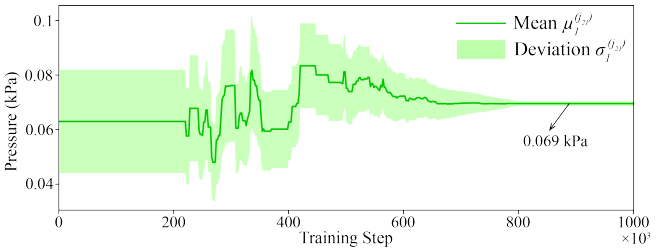
TABLE III: Top performers

Another key observation is the gradual shift of π_d from a generalized controller, effective across a wide set of designs, toward a specialized policy tuned for d^*. This transition is evident in the narrowing gap between the mean reward and the best-performing design (the shaded region in Fig. 2a). Early in training, π_d had to accommodate designs sampled from the entire q_ϕ, resulting in high variance. Later, as q_ϕ concentrated on promising candidates, the variance shrank, indicating policy specialization.

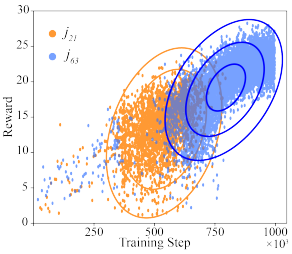
To evaluate the importance of proprioceptive feedback and design diversity, we compared the main framework against



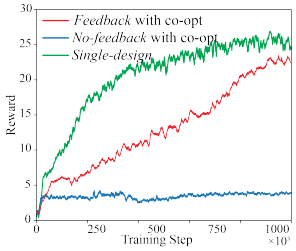
(a) Reward histogram during training, showing the highest scores (blue), running average (red, window = 100), and narrowing gap that reflects increasing specialization of π_d, dashed yellow line explicitly denotes the progress of the discrete set j_{21}.



(b) Convergence of q_ϕ^{cont} to the feasible value.



(c) Reward distributions of the two top-performing nominal structures, j_{21} and j_{63}.

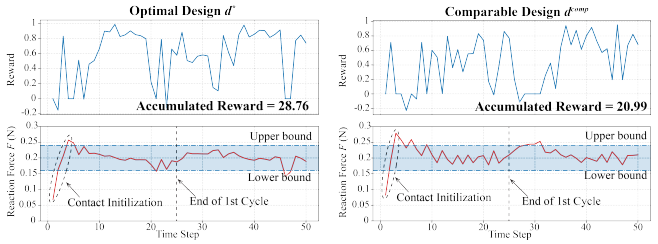


(d) Comparison of training schemes in terms of average reward progression.

Fig. 2: Experiment results.

two baselines: 1) *No-feedback* - where mechanical strain ϵ and morphology information were excluded from the state; 2) Single-design - where training was limited to the final optimal design d^*. All training settings, including network architecture and hyperparameters, remain the same as in our experiment. Figure 3b presents the running-average rewards. In the *No-feedback* case, the policy failed to adapt to different dynamics, producing nearly flat reward curves (blue). The *Single-design* case showed strong improvement (green), even surpassing the full design-space run at intermediate stages, since the controller trained exclusively on one morphology. However, by the end of training, performance converged closely with the main framework, indicating that shared representations in π_d^* allowed knowledge transfer across designs without loss of final performance.

The efficiency of the optimal control–design pair $\{\pi_{d^*}^*, d^*\}$ was validated against d^{comp} and its best controller. Figure 3 plots the resulting reaction forces over 50 testing steps, covering two oscillation cycles of the vibrating plate.



(a) Best design d^* demo. (b) Comparable design d^{comp} demo.

Fig. 3: Performance of optimal design d^* and its comparable contestant d^{comp} under control of $\pi^*_{d^*}$.

Both prototypes required initial contact settling, after which control governed the interaction. Although both designs maintained F within (*i.e.,* 20% of F_t), d^{comp} showed greater fluctuations and drift toward tolerance boundaries. In contrast, $\pi^*_{d^*}$ tightly regulated F within the bounds, even during transitions between vibration cycles. This resilience reflects stronger morphology–control synergy. Overall, the optimal pair $\{\pi^*_{d^*}, d^*\}$ achieved a much higher accumulated reward (28.76 vs. 20.99), confirming its superiority over d^{comp}. The two demonstrations can be reviewed in the Supplementary Video.

VI. Conclusion

This work introduced the first co-design framework for soft robots aimed at executing complex tasks that strongly benefit from feedback-driven control. A central idea was the use of actively variable morphology, which adjusts proprioceptive signals in real time to support optimal and adaptive robot behaviors. By representing these variations as continuous parameters within the design space, we constructed a hybrid formulation: discrete variables define the nominal robot structure, while continuous variables capture its transformable states. Under this paradigm, a single nominal structure can expand its task horizon by switching between morphological states, thereby improving adaptability.

To identify the optimal control–design pair $\{\pi^*_{d^*}, d^*\}$, we trained a design-conditioned control policy π_d with PPO reinforcement learning across the full distribution space. We then proposed an update strategy for both discrete and continuous domains to shift the distribution toward regions of higher reward density. As training progressed, the controller and design co-evolved, converging to an optimal solution without any external supervision. Our evaluations confirmed that the framework consistently discovered the best-performing design–control pair, while training without morphology-driven feedback failed to achieve comparable outcomes. Both quantitative results and behavioral demonstrations validated the effectiveness of the learned $\{\pi^*_{d^*}, d^*\}$.

Despite these contributions, several challenges and open questions remain. First, our results are limited to simulation, and transferring the approach to hardware is a non-trivial step. Issues such as imperfect material modeling, frictional inconsistencies, and other sim-to-real gaps must be addressed before real-world deployment. Second, while our update rule for continuous parameters proved effective here, its scalability to higher-dimensional and wider expand morphologies is uncertain. The covariant nature and nonlinear dynamics of soft bodies may still lead optimization into local minima. More expressive models, such as mixtures of multivariate distributions, may help maintain exploration and avoid premature convergence. Furthermore, we also plan to implement this framework for other soft robotic applications with the primary aim of enhancing their resilience to critical conditions, such as physical damage.

References

[1] R. Pfeifer and G. Gómez, *Morphological Computation – Connecting Brain, Body, and Environment*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 66–83. [Online]. Available: https://doi.org/10.1007/978-3-642-00616-6_5

[2] N. H. Nguyen and V. A. Ho, "Mechanics and morphological compensation strategy for trimmed soft whisker sensor," *Soft Robotics*, vol. 9, no. 1, pp. 135–153, 2022, pMID: 33464996. [Online]. Available: https://doi.org/10.1089/soro.2020.0056

[3] Z. Chen, D. Wu, Q. Guan, D. Hardman, F. Renda, J. Hughes, T. G. Thuruthel, C. D. Santina, B. Mazzolai, H. Zhao, and C. Stefanini, "A survey on soft robot adaptability: Implementations, applications, and prospects [survey]," *IEEE Robotics Automation Magazine*, pp. 2–14, 2025.

[4] D. D. Nguyen, N. H. Nguyen, and V. A. Ho, "Morphology-changeable soft pads facilitate locomotion in wet conditions," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2983–2990, 2023.

[5] T. Chen, Z. He, and M. Ciocarlie, "Co-designing hardware and control for robot hands," *Science Robotics*, vol. 6, no. 54, p. eabg2133, 2021. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abg2133

[6] P. Mannam, X. Liu, D. Zhao, J. Oh, and N. Pollard, "Design and control co-optimization for automated design iteration of dexterous anthropomorphic soft robotic hands," in *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft)*, 2024, pp. 332–339.

[7] C. Schaff, D. Yunis, A. Chakrabarti, and M. R. Walter, "Jointly learning to construct and control agents using deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9798–9805.

[8] R. Koike, R. Ariizumi, and F. Matsuno, "Simultaneous optimization of discrete and continuous parameters defining a robot morphology and controller," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 13 816–13 829, 2024.

[9] K. Sims, *Evolving Virtual Creatures*, 1st ed. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3596711.3596785

[10] D. J. H. III, P. Abbeel, and L. Pinto, "Task-agnostic morphology evolution," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=CGQ6ENUMX6

[11] T. F. Nygaard, D. Howard, and K. Glette, "Real world morphological evolution is feasible," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1392–1394. [Online]. Available: https://doi.org/10.1145/3377929.3398095

[12] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, no. 6799, pp. 974–978, Aug 2000. [Online]. Available: https://doi.org/10.1038/35023115

[13] D. Matthews, A. Spielberg, D. Rus, S. Kriegman, and J. Bongard, "Efficient automatic design of robots," *Proceedings of the National Academy of Sciences*, vol. 120, no. 41, p. e2305180120, 2023. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2305180120

[14] A. Gupta, S. Savarese, S. Ganguli, and L. Fei-Fei, "Embodied intelligence via learning and evolution," *Nature Communications*, vol. 12, no. 1, p. 5721, Oct 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-25874-z

[15] D. Pathak, C. Lu, T. Darrell, P. Isola, and A. A. Efros, *Learning to control self-assembling morphologies: a study of generalization via modularity*. Red Hook, NY, USA: Curran Associates Inc., 2019.
[16] T. Wang, Y. Zhou, S. Fidler, and J. Ba, "Neural graph evolution: Automatic robot design," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=BkgWHnR5tm
[17] D. Ha, "Reinforcement learning for improving agent design," *Artificial Life*, vol. 25, no. 4, pp. 352–365, 2019.
[18] A. Zhao, J. Xu, M. Konaković-Luković, J. Hughes, A. Spielberg, D. Rus, and W. Matusik, "Robogrammar: graph grammar for terrain-optimized robot design," *ACM Trans. Graph.*, vol. 39, no. 6, Nov. 2020. [Online]. Available: https://doi.org/10.1145/3414685.3417831
[19] *Evolutionary Synthesis of Sensing Controllers for Voxel-based Soft Robots*, ser. ALIFE 2022: The 2022 Conference on Artificial Life, vol. ALIFE 2019: The 2019 Conference on Artificial Life, 07 2019. [Online]. Available: https://doi.org/10.1162/isal_a_00223
[20] A. Spielberg, A. Amini, L. Chin, W. Matusik, and D. Rus, "Co-learning of task and sensor placement for soft robotics," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1208–1215, 2021.
[21] F. Corucci, N. Cheney, F. Giorgio-Serchi, J. Bongard, and C. Laschi, "Evolving soft locomotion in aquatic and terrestrial environments: Effects of material properties and environmental transitions," *Soft Robotics*, vol. 5, no. 4, pp. 475–495, 2018, pMID: 29985740. [Online]. Available: https://doi.org/10.1089/soro.2017.0055
[22] S. Kriegman, S. Walker, D. S. Shah, M. Levin, R. Kramer-Bottiglio, and J. Bongard, "Automated shapeshifting for function recovery in damaged robots," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.
[23] J. S. Bhatia, H. Jackson, Y. Tian, J. Xu, and W. Matusik, "Evolution gym: a large-scale benchmark for evolving soft robots," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2021.
[24] T. Wang, P. Ma, A. E. Spielberg, Z. Xian, H. Zhang, J. B. Tenenbaum, D. Rus, and C. Gan, "Softzoo: A soft robot co-design benchmark for locomotion in diverse environments," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/forum?id=Xyme9p1rpZw
[25] C. Schaff, A. Sedal, and M. Walter, "Soft Robots Learn to Crawl: Jointly Optimizing Design and Control with Sim-to-Real Transfer," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.
[26] C. Schaff, A. Sedal, S. Ni, and M. R. Walter, "Sim-to-real transfer of co-optimized soft robot crawlers," *Auton. Robots*, vol. 47, no. 8, p. 1195–1211, Sep. 2023. [Online]. Available: https://doi.org/10.1007/s10514-023-10130-8
[27] N. H. Nguyen and V. A. Ho, "Tactile compensation for artificial whiskered sensor system under critical change in morphology," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3381–3388, 2021.
[28] N. H. Nguyen, H. Hauser, P. Maiolino, and V. A. Ho, "Tactile resilience of sensory whisker by adaptive morphology," *IEEE Access*, vol. 10, pp. 101 814–101 824, 2022.
[29] Q. K. Luu, N. H. Nguyen, and V. A. Ho, "Simulation, learning, and application of vision-based tactile sensing at large scale," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2003–2019, 2023.
[30] B. G. Cangan, S. E. Navarro, B. Yang, Y. Zhang, C. Duriez, and R. K. Katzschmann, "Model-based disturbance estimation for a fiber-reinforced soft manipulator using orientation sensing," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 9424–9430.
[31] P. Schegg, E. Ménager, E. Khairallah, D. Marchal, J. Dequidt, P. Preux, and C. Duriez, "Sofagym: An open platform for reinforcement learning based on soft robot simulations," *Soft Robotics*, 2022.