Pretraining-Finetuning Framework for Efficient Codesign of Legged Robots

Ci Chen, Yifei Yu, Haoqi Han, Yue Wang, *Member, IEEE*, Rong Xiong, *Senior Member, IEEE*, and Hesheng Wang, *Senior Member, IEEE*

***Abstract*—In nature, animals with exceptional locomotion abilities, such as cougars, often exhibit asymmetry between their forelimbs and hindlimbs. This observation inspires our investigation into whether optimizing leg length in legged robots can impart similar locomotion capabilities. In this article, we propose a method that cooptimizes mechanical structure and control policies to enhance the locomotion performance of legged robots. Specifically, we introduce a pretraining-finetuning framework that not only ensures optimal control strategies for each mechanical configuration but also enhances time efficiency. Furthermore, we develop an innovative training approach for our pretraining network that integrates spatial domain randomization with discount regularization techniques, significantly improving the network's generalizability. The effectiveness of the proposed method is validated through two quadrupedal locomotion tasks and two parkour-style tasks. Experimental results demonstrate that the pretraining-finetuning framework markedly enhances overall codesign performance with reduced time expenditure. Additionally, experimental validation is performed, confirming that the cooptimized morphology and control strategy effectively mitigate the robot's energy expenditure during designated tasks.**

***Index Terms*—Codesign, deep reinforcement learning (DRL), legged robot.**

Received 18 August 2025; revised 25 November 2025; accepted 2 January 2026. This work was supported in part by the National Key R&D Program of China under Grant 2024YFB4708900; in part by the Natural Science Foundation of China under Grant 62225309, Grant U24A20278, Grant 62361166632, and Grant 62503316; and in part by the China Postdoctoral Science Foundation under Grant 2025M771686. *(Corresponding author: Hesheng Wang.)*

Ci Chen, Haoqi Han, and Hesheng Wang are with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, the State Key Laboratory of Avionics Integration and Aviation System-of-Systems Synthesis, and Shanghai Key Laboratory of Navigation and Location Based Services, Shanghai 200240, China (e-mail: chenci107@sjtu.edu.cn; hhq123@sjtu.edu.cn; wanghesheng@sjtu.edu.cn).

Yifei Yu is with the College of Automation Engineering, Shanghai University of Electric Power, Shanghai 201306, China (e-mail: yuyifei@mail.shiep.edu.cn).

Yue Wang and Rong Xiong are with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China (e-mail: wangyue@iipc.zju.edu.cn; rxiong@zju.edu.cn).

Digital Object Identifier 10.1109/TIE.2026.3654778

I. Introduction

QUADRUPEDAL robots, capable of utilizing independent support points to provide stable support and traction, are inherently well suited for diverse terrains and have garnered significant attention in recent years [1], [2], [3], [4]. Currently, the design of leg lengths in common quadrupedal robots largely relies on the experiential knowledge of mechanical engineers, lacking a clear theoretical foundation [5]. However, the locomotion performance of quadrupedal robots is constrained not only by control strategies but also by their mechanical structures, highlighting a complex interplay between the two. Exploring the codesign of structural and control parameters is essential to significantly enhance the robot's ability to adapt to complex environments, a task often referred to as "codesign" [5], [6], [7], [8], [9].

The codesign of robots can be primarily divided into methods based on theoretical analysis [10], [11] and methods based on data-driven [5], [6], [7], [8], [9], [12], [13], [14]. The theoretical analysis-based codesign methods typically refer to optimal control strategies based on robotic dynamics models, which are relatively intuitive. Ha et al. [10] modeled the robot's motion as a spatiotemporal solution to the optimal control problem and established a complex relationship between structural and motion parameters through sensitivity analysis. This relationship between structure and performance can guide the automated design of robotic structures. Research on a small quadrupedal robot revealed that utilizing this optimization method resulted in a 32% reduction in maximum torque compared to the original design. With the widespread use of whole-body control (WBC) in the field of quadrupedal robot control, De et al. [15] incorporated WBC into the structural parameter optimization process, utilizing gradient methods to adjust any continuous variables relied upon during the robot's design process. Additionally, this approach derived precise analytical expressions for the required derivatives through sensitivity analysis. Although the aforementioned studies offer theoretical assurances, they face significant limitations in applications. First, such methods often rely on simplified models to reduce complexity, such as trajectory optimization focusing on center-of-mass dynamics while neglecting limb mass, causing structural parameter optimization to deviate from actual dynamics and reducing real-world applicability. Second, trajectory optimization results depend on manually

defined base actions and predefined gait patterns, limiting adaptability to varying operational contexts.

With the widespread application of data-driven methods in the field of motion control for legged robots [1], [2], the use of data-driven approaches for codesign has emerged as a viable technical pathway. Data-driven methods, particularly those based on deep reinforcement learning (DRL), are well suited for determining control strategies under specified structural parameters, as they facilitate the acquisition of control policies without the need for model simplification or heuristic assumptions. Such methods typically frame the problem as a bilevel optimization issue, with the lower level optimizing the control strategy for a specified morphology and the upper level searching for the best morphology given the optimal control policy. Depending on how the control strategy is obtained, there are mainly two categories. The first involves training specific controllers for given candidates during the morphology optimization, as shown in Fig. 1(a). For instance, Gupta et al. [13] used a tournament algorithm at the upper level, employing the fitness of each candidate morphology as input to derive the optimal morphology. On the lower level, it utilizes 1152 CPUs with DRL algorithms to learn the optimal control policies for various candidate morphologies, which are highly time-consuming and require substantial computational power. In contrast, the second category focuses on the development of a versatile and generalizable model [6], [8], [9], which is capable of providing alternative strategies applicable to various morphologies, as illustrated in Fig. 1(b). Specifically, Yu et al. [8] proposed the embodiment-aware transformer (EAT), an architecture that transforms control problems into conditional sequence modeling tasks. EAT utilizes a causal mask to produce optimal actions. By conditioning the autoregressive model on the desired robot configuration, past states, and actions, the model is able to generate future actions that best align with the current robot structure. This method also demonstrated that optimization of structural parameters alone could enable the robot to step over obstacles of 10 cm in height. Similarly, Chen et al. [6] employed a concurrent network architecture, while Bjelonic et al. [9] utilized a teacher–student architecture to train a general model serving as a surrogate for control policies of robots with different morphologies. However, this approach of substituting optimal strategies with generalized ones, while time-saving, cannot ensure optimality in specific morphologies, potentially impairing the efficacy of upper level optimization decisions.

To maintain optimality while also conserving computational power, in this article, we introduced a pretraining-finetuning framework for quadruped robots' codesign, with the goal of improving their locomotion performance, as shown in Fig. 1(c). We first trained a locomotion strategy that is adaptable across a variety of morphologies. Specifically, we integrated spatial domain randomization and discount regularization techniques. The former enables simultaneous training of thousands of robots with varying morphologies, while the latter helps in reducing memory biases in the value network. The combination of them significantly enhances the generalizability of the pretrained model. In the subsequent phase of morphology optimization, we fine-tuned this generalized model to suit each specific morphology candidate. Our experimental results indicate that merely 400 steps of fine-tuning (6.67% of the typical training steps) were sufficient for the new model to converge, delivering performance comparable to models specifically designed for each structure. In summary, the contributions of this article are as follows.

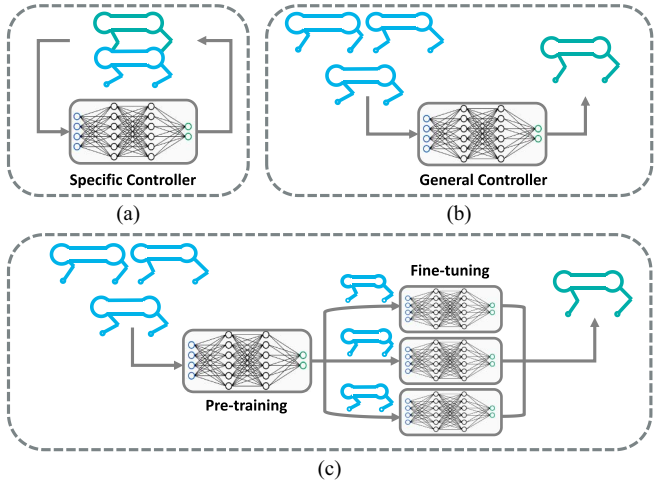


Fig. 1. Comparison of codesign methods. (a) Designing distinct control strategies tailored to various morphologies. (b) Utilizing a generalized strategy as a substitute. (c) Algorithm we introduce starts by training a generalized model across varying morphologies, followed by fine-tuning for a specific morphology.

1) We introduced a pretraining-finetuning framework to address the codesign challenge in robotics, ensuring optimality while conserving computational resources.
2) We proposed the concept of combining spatial domain randomization with discount regularization to obtain a pretrained model, effectively enabling generalization across robots with different morphologies.
3) Extensive ablation and comparative experiments have validated the efficacy of our proposed algorithm in codesign tasks, demonstrating its capability to improve the locomotion performance of robots.

The remainder of this article is organized as follows. Section II models the codesign problem. Section III delineates the proposed methodologies. Section IV presents an evaluation and analysis of the experimental results, and Section V concludes the discussion.

II. Problem Formulation

Codesign for robots can be formulated as a bilevel optimization problem

$$\max_{\xi\in\Xi} \mathcal{F}(\pi^*(\xi),\xi)$$
$$\text{s.t. } \pi^*(\xi) = \arg\max_{\pi\in\Pi} \mathcal{J}(\pi,\xi) \tag{1}$$

where π is the control policy, and ξ is the morphology parameters. $\mathcal{F}(\cdot)$ and $\mathcal{J}(\cdot)$ are objective functions of the upper and lower layers, respectively. Lower level optimization is

performed first to obtain optimized policies under predefined morphology parameters. Then, the upper level optimization is performed to obtain the optimized morphology among the morphology parameter search space based on the fitness acquired by the optimized policies.

The interaction between the robot and its environment can be modeled as an extension of the Markov decision process (MDP) conditioned by ξ. This can be represented by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action space, $\mathcal{P}$ and $\mathcal{R}$ denote the dynamics and reward function, and $\gamma \in [0, 1)$ indicates the discount factor. At each time step t, an agent selects an action $a_t \in \mathcal{A}$ under the state $s_t \in \mathcal{S}$ according to the policy $\pi(\cdot|s_t, \xi)$ and receives a reward $r_t = r(s_t, a_t, \xi)$. The environment then transitions to a new state s_{t+1} following the transition model $p(s_{t+1}|s_t, a_t, \xi)$. The objective of lower level optimization is to optimize the control policy to maximize the expectation of the cumulative rewards conditioned on a specific morphology parameter $\bar{\xi}$. The evaluation is depicted as follows:

$$\mathcal{J}\left(\pi, \bar{\xi}\right) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r\left(s_t, a_t, \bar{\xi}\right) \middle| a_t \sim \pi\left(\cdot \middle| s_t, \bar{\xi}\right)\right] \tag{2}$$

where T refers to the horizon. In (2), ξ is fixed and π is to be optimized. Similarly, the objective of the upper level is to find the optimized morphology parameter ξ^* that maximizes the evaluation function given the optimal policy $\pi^*(\xi)$. The evaluation function is described as follows:

$$\mathcal{F}\left(\pi^*\left(\xi\right), \xi\right) = \mathbb{E}_{s_t \sim \tau(\pi^*(\xi))}\left[f(s_t, \xi)\right] \tag{3}$$

where $f(s_t, \xi)$ serves as the evaluation function, which can be designed in different forms according to different objectives, while $\tau(\pi^*(\xi))$ represents the trajectories generated using the $\pi^*(\xi)$ policy. Here, π remains fixed, and ξ is the parameter subject to optimization.

Obtaining $\pi^*(\xi)$ in (3) is time-consuming. To address this issue, some methods [6], [8], [9] use a general policy instead of specific strategies for each morphology, where the general policy is constructed from historical interaction data

$$\mathcal{F}\left(\pi^*\left(\xi\right), \xi\right) \approx \mathcal{F}\left(\pi_{\text{gen}}\left(\xi\right), \xi\right) = \mathbb{E}_{s_t \sim \tau(\pi_{\text{gen}}(\xi))}\left[f(s_t, \xi)\right]. \tag{4}$$

This approach reduces the algorithm's complexity from $O(N)$ to $O(1)$. However, it fails to ensure the optimality of the control policy for specific structural parameters, resulting in inaccurate fitness assessments and affecting the effectiveness of codesign. Therefore, we propose a pretraining and fine-tuning framework, where the fine-tuned policy for specific configurations is used as the alternative strategy in the upper level optimization process

$$\mathcal{F}\left(\pi^*\left(\xi\right), \xi\right) \approx \mathcal{F}\left(\pi_{\text{fine}}\left(\xi\right), \xi\right) = \mathbb{E}_{s_t \sim \tau(\pi_{\text{fine}}(\xi))}\left[f(s_t, \xi)\right]. \tag{5}$$

In subsequent experiments, it can be found that for the specific morphology ξ, the final performance obtained with the fine-tuned policy π_{fine} is very close to that of π^*. Thus, compared to π_{gen}, π_{fine} can be regarded as a superior alternative strategy.

III. Methodology

A. System Overview

The overall framework is illustrated in Fig. 3, with the grey section indicating the training phase and the green section the deployment phase. The training phase comprises the pretraining phase (Section III-B) and the Bayesian optimization (BO) phase with embedded fine-tuning (Section III-C). During the pretraining phase, we develop a control policy that generalizes to varying morphologies, serving as the initial model. In the subsequent fine-tuning phase, the algorithm iteratively identifies the optimal morphology using the fitness of candidate morphologies as prior knowledge. Since the pretrained model might not be optimal for every specific morphology, it undergoes fine-tuning to enhance accuracy. This fine-tuning enables more precise fitness calculations within the BO process, improving the codesign framework's overall effectiveness. For the deployment phase (Section III-D), we reconfigure the robot according to the morphology parameters obtained in the fine-tuning phase and acquire an executable policy via supervised learning.

1) State and Action Space: The state consists of five components: proprioception x_t, explicit privileged state e'_t, implicit privileged state e_t, terrain elevation samples m_t, and historical proprioception h_t. x_t contains body angular velocity, orientation angles, yaw deviation, joint states, and the previous action. e_t includes the robot's mass, center-of-mass position, structural parameters, foot-terrain friction coefficient, and motor damping ratio. e'_t represents body linear velocity, while h_t maintains a temporal sequence of proprioception. The action a_t represents the target joint angles for the robot's 12 joints. These target angles are converted into torques using a proportional–derivative (PD) controller and then sent to the robot.

2) Reward Functions: For the quadruped locomotion task, the main reward is designed to encourage the robot to track a commanded linear velocity. For the quadruped parkour task, we follow the approach from Cheng et al. [17] and set "goal tracking" as the primary reward term. This encourages the robot to move toward waypoints and incentivizes it to jump onto high platforms rather than circumvent them. The specific equations and weights of each reward term are detailed in Table I.

B. Pretraining Phase

1) Spatial Domain Randomization: Traditionally, domain randomization involves altering variables such as friction coefficients and motor strengths to bridge the sim-to-real gap [18], a process we refer to as temporal domain randomization (Temporal-DR). In this work, we adapt this core idea for a different purpose: to train a single control policy that can generalize across a variety of robot morphologies within the simulation. A naive temporal approach to this morphological randomization would involve sequentially loading robots with different structural parameters. However, since these parameters are fixed at initialization, such an approach would require repeatedly reloading the simulator, significantly slowing down the training process. To address this

TABLE I
REWARD TERMS FOR QUADRUPED LOCOMOTION AND PARKOUR TASKS

Quadruped Locomotion			Quadruped Parkour		
Reward	Equation	Weight	Reward	Equation	Weight
Linear velocity tracking	$\exp(-\|\boldsymbol{v}_{xy}-\boldsymbol{v}_{xy}^{\mathrm{cmd}}\|_1)$	1.0	Goal tracking	$\min(\langle \boldsymbol{v}, \hat{\boldsymbol{d}}_w\rangle, v^{\mathrm{cmd}})$ $\hat{\boldsymbol{d}}_w=\frac{\boldsymbol{p}-\boldsymbol{x}}{\|\boldsymbol{p}-\boldsymbol{x}\|}$	1.5
Collisions	$\sum_{i=0}^{12} c_i \cdot M_{\mathrm{body}}(\boldsymbol{p}_i)$	-1.0	Clearance	$-\sum_{f=0}^{3} c_f \cdot M_{\mathrm{foot}}(\boldsymbol{p}_f)$	-1.0
Angular velocity tracking	$\exp(-\|\boldsymbol{\omega}_z-\boldsymbol{\omega}_z^{\mathrm{cmd}}\|_1)$	0.5	Angular velocity tracking	$\exp(-\|\boldsymbol{\omega}_z-\boldsymbol{\omega}_z^{\mathrm{cmd}}\|_1)$	0.5
Linear velocity penalty	$\boldsymbol{v}_z^2$	-2.0	Linear velocity penalty	$\boldsymbol{v}_z^2$	-1.0
Angular velocity penalty	$\boldsymbol{\omega}_{xy}^2$	-0.05	Angular velocity penalty	$\boldsymbol{\omega}_{xy}^2$	-0.05
Action rate	$\|\boldsymbol{a}_t-\boldsymbol{a}_{t-1}\|_2$	-0.01	Action rate	$\|\boldsymbol{a}_t-\boldsymbol{a}_{t-1}\|_2$	-0.1
Hip position penalty	$(\boldsymbol{q}_{\mathrm{hip}}-\boldsymbol{q}_{\mathrm{hip}}^{\mathrm{init}})^2$	-0.5	Hip position penalty	$(\boldsymbol{q}_{\mathrm{hip}}-\boldsymbol{q}_{\mathrm{hip}}^{\mathrm{init}})^2$	-0.5
Joint acceleration	$\ddot{\boldsymbol{q}}^2$	-2.5×10^{-7}	Joint acceleration	$\ddot{\boldsymbol{q}}^2$	-2.5×10^{-7}
Joint position penalty	$(\boldsymbol{q}-\boldsymbol{q}^{\mathrm{init}})^2$	-0.04	Joint position penalty	$(\boldsymbol{q}-\boldsymbol{q}^{\mathrm{init}})^2$	-0.04
Torque rate	$(\boldsymbol{\tau}-\boldsymbol{\tau}_{t-1})^2$	-1×10^{-7}	Torque rate	$(\boldsymbol{\tau}-\boldsymbol{\tau}_{t-1})^2$	-1×10^{-7}
Torque penalty	$\boldsymbol{\tau}_t^2$	-1×10^{-5}	Torque penalty	$\boldsymbol{\tau}_t^2$	-1×10^{-5}
Feet air time	$\sum_{f=0}^{3}(t_{\mathrm{air},f}-0.5)$	1.0	Orientation	$\boldsymbol{g}^2$	-1.0

Note: In this formulation, bold symbols denote vectors. The robot's position is $\boldsymbol{x}$ and the next waypoint is $\boldsymbol{p}$. Superscripts $(\cdot)^{\mathrm{cmd}}$ and $(\cdot)^{\mathrm{init}}$ represent commanded and initial values, respectively. For instance, v^{cmd} is the commanded speed, while $\boldsymbol{q}^{\mathrm{init}}$ is the initial joint configuration. The coefficient c_i (or c_f) is a binary indicator, set to 1 to flag a collision for the i-th body link or to select the f-th foot for evaluation. The robot's base state is described by its linear velocity $\boldsymbol{v}\in\mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}\in\mathbb{R}^3$. The control action is $\boldsymbol{a}_t\in\mathbb{R}^{12}$. Joint-related variables include angles $\boldsymbol{q}\in\mathbb{R}^{12}$, accelerations $\ddot{\boldsymbol{q}}\in\mathbb{R}^{12}$, and torques $\boldsymbol{\tau}\in\mathbb{R}^{12}$. For the locomotion task, M_{body} checks for collisions, and $t_{\mathrm{air},f}$ is the air time for foot f. For the parkour task, M_{foot} is a Boolean function that returns 1 if foot $\boldsymbol{p}_f$ is within 5cm of an edge, and $\boldsymbol{g}$ is the gravity vector projected onto the robot's body frame.

computational challenge and achieve efficient morphological generalization, we introduce the concept of spatial domain randomization (Spatial-DR). Leveraging the extensive parallel capabilities of Isaac Gym [19], we implement this approach by simultaneously training thousands of robots with varied morphologies within a single simulation instance.

In alignment with our previous work [7], we generate robots with varying morphological parameters by modifying the unified robot description format (URDF) files of quadruped robots. The URDF representation of a quadruped robot is depicted in Fig. 2(a). While our methodology is applicable to a wide range of structural parameters, we focus specifically on the thigh and shank lengths for a more focused analysis of the optimization outcomes. Adjustments are required for the *inertial*, *visual*, and *collision* attributes of the corresponding links. Additionally, modifications to the *origin* parameters of the knee and ankle joints are necessary, with detailed procedures provided in Table II. To maintain symmetry and ensure the robot's stability, we impose symmetry between the corresponding parts of the left and right legs. Consequently, the adjustable parameters are the scaling factors for the front thighs, front shanks, hind thighs, and hind shanks. During the training of the generalizable model, we allow the scaling factors to be randomly sampled from a feasible range, $\xi_l \sim \mathcal{U}(c_{\min}, c_{\max})$. The sampled parameters, ξ_l, are then used to construct robots with varying leg lengths, as shown in Fig. 2(b).

2) Discount Regularizations: The proposed spatial domain randomization method enables the algorithm to train using data

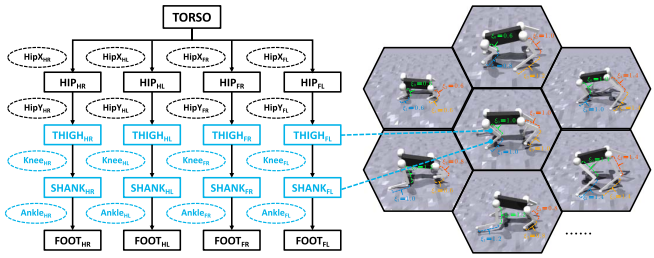


Fig. 2. Schematic representation of the robot's URDF model, where solid rectangles denote links and dashed ellipses indicate joints. The black components represent fixed parameters, while the blue components signify adjustable parameters. (a) URDF structure of the quadruped robot. (b) Quadruped robots with different morphologies.

from robots with varying morphologies. However, as introduced in [20] and [21], a value network trained across multiple agents is more likely to memorize the training data, leading to poor generalization to unvisited states. This not only hinders training performance but also negatively impacts testing performance in new environments. To mitigate this issue, we employed a *discount regularization* approach, which adjusts the discount factor to a smaller value, denoted as γ_{reg}, where $0<\gamma_{\mathrm{reg}}<\gamma<1$. The modification of the discount factor influences the calculation of the advantage function and temporal difference in the PPO algorithm, thereby affecting the computation of the actor and critic loss functions. The new formulations for the

TABLE II
THE PARAMETERS THAT NEED TO BE MODIFIED IN THE URDF FILE

Tag	Attribute	Parameter	Value
Link	Inertial	Origin	x:0, y: 0, z: $\frac{-L_l \times \xi_l}{2}$
		Mass	value: $m_l \times \xi_l$
		Inertia	ixx: $\frac{m_l \xi_l}{12}\left(b_l^2 + (L_l \xi_l)^2\right)$ iyy: $\frac{m_l \xi_l}{12}\left(b_l^2 + (L_l \xi_l)^2\right)$ izz: $\frac{m_l \xi_l b_l^2}{6}$
	Visual	Origin	x:0, y:0, z: $\frac{-L_l \times \xi_l}{2}$
		Geometry/Box	size: $b_l, b_l, L_l \times \xi_l$
	Collision	Origin	x:0, y:0, z: $\frac{-L_l \times \xi_l}{2}$
		Geometry/Box	size: $b_l, b_l, L_l \times \xi_l$
Knee joint	/	Origin	x:0, y:0, z: $z_{\text{knee}} \times \xi_{l,l=0,2}$
Ankle joint	/	Origin	x:0, y:0, z:$z_{\text{ankle}} \times \xi_{l,l=1,3}$

Note: The index l represents different components of the robot, with $l = 0$ corresponding to front thighs, $l = 1$ to front shanks, $l = 2$ to hind thighs, and $l = 3$ to hind shanks. The variable L_l represents the original length of each leg, while m_l signifies the original mass of each leg. The collision model of the leg is represented by a cuboid with a square cross-section, where b_l denotes the side length of the square cross-section. Furthermore, z_{knee} represents the original position of the knee joint, and z_{ankle} denotes the original position of the ankle joint.

advantage function and temporal difference are as follows:

$$A_t^{\gamma_{\text{reg}}} = \sum_{l=0}^{\infty} (\gamma_{\text{reg}}\lambda)^l \delta_{t+l}^{\gamma_{\text{reg}}} \tag{6}$$

$$\delta_t^{\gamma_{\text{reg}}} = r_t + \gamma_{\text{reg}} V_\phi(s_{t+1}) - V_\phi(s_t) \tag{7}$$

where λ is the hyperparameter used in generalized advantage estimation (GAE) [22]. The discount factor γ_{reg} helps to stabilize training by setting the agent's planning horizon. A high γ_{reg} aims for the best long-term rewards but risks creating high variance and unstable learning. Lowering γ_{reg} avoids this by making the agent focus on immediate rewards. This approach biases the agent toward short-term success, but, in return, it significantly reduces variance and leads to more reliable training. In short, trading the best possible outcome for more stable progress is an effective regularization technique. The pseudocode for the pretraining process is presented in Algorithm 1.

Algorithm 1: Pre-training Phase.

Given: Initial policy parameters θ_0, initial value function parameters ϕ_0.

1: $\xi_l \sim \mathcal{U}(c_{\min}, c_{\max})$ for $l = 0, 1, 2, 3$;
2: Create N_{pre} robots with different morphologies parameterized by ξ_l;
3: **for** $k = 1$ **to** K **do**
4: Collect trajectories $D_k = \{\tau_k\}$ by running policy $\pi_k = \pi(\theta_k)$;
5: Compute advantage estimates $A_t^{\gamma_{\text{reg}}}$ via Eq. (6);
6: Update policy by maximizing PPO objective with $A_t^{\gamma_{\text{reg}}}$;
7: Update value function by minimizing $\delta_t^{\gamma_{\text{reg}}}$ via Eq. (7);
8: **end for**
9: **return** Pre-trained policy $\pi_{\theta_{\text{pre}}}$.

C. Fine-Tuning Phase

1) Fine-Tuning Based on Pretrained Model: By integrating spatial domain randomization with discount regularization, we have developed a control strategy applicable to robots with varying morphologies. However, this strategy does not guarantee optimality in each morphology. To provide a more reliable fitness function for the morphology optimization algorithm, each morphological candidate is fine-tuned. Unlike the training of generalization models, we removed the spatial domain randomization method during the fine-tuning process. Instead, we created thousands of identical robots to interact with the environment, allowing us to fine-tune specifically for the targeted morphology.

2) Morphology Optimization: The objective of morphology optimization is to find optimal morphology parameters $\xi \in \Xi$ given control strategy $\pi^* \in \Pi$, such that the evaluation function $\mathcal{F}(\pi^*(\xi), \xi) : \Xi \times \Pi \to \mathbb{R}$ is maximized. The optimization objective (or *fitness*) $f(\cdot)$ can be formulated in various ways depending on the specific requirements of the task. In our previous work [6], we demonstrated that the BO method is well suited for optimizing morphology parameters in codesign, given that ξ is nondifferentiable with respect to $f(\cdot)$, we can summarize the optimization problem as follows:

$$\begin{aligned} \xi^* &= \arg\max_{\xi \in \Xi} \mathcal{F}(\pi_{\text{fine}}(\xi), \xi) \\ &= \arg\max_{\xi \in \Xi} \mathbb{E}_{s_t \sim \tau(\pi_{\text{fine}}(\xi))} \left[f(s_t, \xi) \right]. \end{aligned} \tag{8}$$

The BO algorithm is an iterative process, as shown in Algorithm 2. In each iteration, a candidate morphology ξ^i is selected. Based on this chosen morphology, the lower level algorithm initially fine-tuned the control policy on the basis of the pretrained model. After deriving the strategy $\pi_{\text{fine}}(\xi^i)$, the robot equipped with the morphology parameters ξ^i interacts with the environment under the guidance of $\pi_{\text{fine}}(\xi^i)$ to obtain $f(\cdot)$. This outcome is then fed back to the BO algorithm, which continues to iterate until a specified number of rounds are completed, ultimately yielding the optimal morphology ξ^*.

In our simulation experiments, the optimization objective $f(\cdot)$ is defined as the robot's nondiscounted cumulative rewards within a single epoch. This metric is chosen as it offers a comprehensive evaluation of the robot's overall performance. The objective function is formulated as follows:

$$f(\cdot)_{\text{sim}} = \mathbb{E}_N \left(\sum_{t=0}^{T} r(s_t^n, a_t^n, \xi) \right) \tag{9}$$

where T denotes the number of steps in an episode, and N represents the number of robots interacting with the environment. For the physical experiments, however, obtaining precise reward signals as in simulation is not feasible. Consequently, the optimization objective is shifted to the minimization of

Algorithm 2: Fine-tuning Phase.

Given: Pre-trained policy $\pi_{\theta_{\text{pre}}}$.

for $i = 1$ **to** I_{total} **do**
  Create N_{fine} robots with identical morphology parameterized by $\xi^i_{l,l=0,1,2,3}$ (denote it by ξ^i);
  Initialize policy: $\theta_0 \leftarrow \theta_{\text{pre}}$;
  for $k = 1$ **to** K **do**
    Collect trajectories $D_k = \{\tau_k\}$ by running policy $\pi_k = \pi(\theta_k)$;
    Update policy via gradient descent:

$$\theta_{k+1} \leftarrow \theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_{\xi^i}(\pi_{\theta_k})$$

  end for
  Calculate $\mathcal{F}(\pi_{\text{fine}}(\xi^i), \xi^i)$ via Eq. (9) or Eq. (10);
  Augment $D_{BO}^{1:i} = \left\{D_{BO}^{1:i-1}, \left(\xi^i, \mathcal{F}\left(\pi_{\text{fine}}\left(\xi^i\right), \xi^i\right)\right)\right\}$ and update the Gaussian process in BO;
end for
Obtain optimal morphology ξ^* via BO;
return Optimized morphology parameters ξ^*.

energy consumption [5]. This is formulated as a cost function representing the total positive mechanical power, defined as

$$f(\cdot)_{\text{real}} = -\sum_{t=0}^{T}\left(\sum_{j=1}^{12}\max\left(\left(\omega_j \tau_j\right)_t, 0.0\right)\right) \tag{10}$$

where τ_j and ω_j denote the torque and angular velocity of the jth joint, respectively.

D. Deployment Phase

For deployment on a physical robot that relies solely on onboard sensors like proprioception and a forward-facing depth camera, we employ a teacher–student framework. This framework is designed to distill the capabilities of a privileged teacher policy into a student policy that operates from visual input. The student policy's architecture includes several modules prepared in a preliminary stage. First, an estimator network is trained via supervised learning to output the explicit privileged state e'_t from x_t using ground truth from the simulation. Second, a history encoder E_{history} is trained using the regularized online adaptation (ROA) method [23]. Subsequently, the student's core modules, which are the depth encoder E_{depth} and the actor-backbone, are optimized end-to-end with a shared optimizer. This optimization is guided by a dual objective. The first objective is to facilitate motor skill transfer via behavioral cloning by minimizing the MSE between student and teacher actions. The second is to train a dedicated head in the E_{depth} network to regress the heading direction from visual input, using the teacher's oracle heading vector as the ground truth. However, directly using the student's own unreliable heading predictions can induce catastrophic distribution drift. To overcome this, we use the mixture of teacher and student (MTS) strategy [17], [24]. Specifically, the student's predicted heading is used as input to its actor-backbone only when it is sufficiently close to the teacher's oracle heading. Otherwise, the system defaults to the Oracle heading for guidance. This mechanism ensures the student receives reliable navigation commands throughout the learning process, which prevents training collapse and significantly stabilizes knowledge distillation. Furthermore, the optimized morphology for the deployment phase is derived from the output of the BO algorithm.

IV. Experiments

A. Experimental Setup

1) Design Parameters: Our objective is to optimize the design of robot legs. The design parameters serve as scaling factors for the nominal link lengths. During the pretraining phase, we set the number of robots for parallel training as N_{pre}. As detailed in Section III-B1, we initially select the morphological parameters ξ_l [defined in Fig. 2(b)] from the uniform distribution $\mathcal{U}(0.6, 1.4)$ during simulator initialization. After generating N_{pre} parameter sets, such as $[\xi_0, \xi_1, \xi_2, \xi_3]$, we use these sets to generate N_{pre} distinct robot morphologies, which are subsequently loaded into the simulator. During the fine-tuning phase, for each set of candidate design parameters in the BO iteration, N_{fine} identical robots are created to derive the optimal policy for that morphology.

2) Training Details: We utilized Isaac Gym as the simulator, implementing the motion control strategy based on the PPO algorithm [16] using the PyTorch library. In line with the majority of existing studies [17], [25], the policy outputs joint angles, which are subsequently converted into torques using a PD controller. This hierarchical approach is advantageous as it simplifies the learning task by constraining the action space, provides inherent stability through the controller's damping, and enhances robustness for sim-to-real transfer. When robot limb lengths change, PD parameters require adjustment, we apply a correction factor η_j to the corresponding joints' PD parameters via a polynomial from [26]. The initial proportional gain k_p is set to 40, while the derivative gain k_d is established at 0.7. To optimize the design parameters, we leverage the Hyperopt library for hyperparameter tuning over more than 30 generations. All experiments are conducted across three independent runs. We evaluate our method on two benchmark tasks: locomotion and parkour. The locomotion task consists of two terrain environments, namely steps and hills, while the parkour task involves environments designed for high jump and long jump maneuvers. To demonstrate the universality of the proposed algorithm, the initial robot morphology for the locomotion task is derived from Unitree Robotics' Go2 quadruped robot, while that for the parkour task originates from Deeper Robotics' Jueying Lite3 robot.

B. Simulation Results

1) Comparison of Domain Randomization: We design three experiments to demonstrate the spatial domain randomization method's effectiveness in enhancing pretrained models' generalization. Each experiment's design is as follows.

a) *Temporal-DR*: This method originally, proposed in [27], randomizes robot morphology during training to

develop generalized control policies. For our study, 81 distinct morphologies were uniformly sampled. Training progressed sequentially: each morphology set was trained for 1/81 of the total duration before switching to the next, continuing until completion.

b) *No-DR*: Without employing the domain randomization method, all robots used the default morphology during training, with all scaling factors set to one.

c) *Spatial-DR (Ours)*: Our method generates N distinct URDF files as robotic morphologies using Section III-B1 approach, then leverages Isaac Gym's massive parallelism to load and train them concurrently.

We evaluate three domain randomization methods across two tasks, as illustrated in Fig. 4(a). For each method, we randomly sample 100 sets of morphological parameters and compute the mean reward values achieved by the corresponding control policies. The results, including the mean and standard deviation, are reported in Fig. 4(a). Additionally, T-tests are performed to compare the proposed method with Temporal-DR and No-DR, with the results also shown in Fig. 4(b). The results clearly demonstrate that the proposed method achieves the highest mean reward values, with p-values for comparisons against both baselines falling below the 0.05 threshold, confirming its statistical superiority over the baselines.

Furthermore, we evaluated the training time required by each method on a consistent hardware platform. We conducted three experimental trials for each approach. The No-DR method took approximately 2 h 16 min to train. Spatial-DR required a similar average training time of 2 h 18 min, whereas Temporal-DR took nearly twice as long, at approximately 4 h 18 min. The variance across the three trials for each method did not exceed 5 min, indicating consistent performance. The reason Temporal-DR's training time is roughly double that of the other two methods is that it involves the continuous termination of the current environment and the loading of new morphologies throughout the training process. These results suggest that the Spatial-DR method effectively enhances the generalizability of the control policy without imposing significant additional computational overhead.

2) Comparison of Regularization: To verify the effectiveness of regularization methods, we designed the following experiments.

a) *Discount regularization*: The proposed method altered the discount factor γ in the PPO algorithm to a smaller value γ_{reg}. After numerous tests, we settled on $\gamma_{\text{reg}} = 0.98$.

b) *Activation regularization*: A regularization term is incorporated into the objective function of the critic to penalize excessively high value estimates.

c) *Normal*: This approach does not employ regularization.

d) *Phasic policy gradient (PPG) [21]*: A SOTA framework improving PPO's generalization via auxiliary losses during training, details can be found in [21].

A comparative analysis of cumulative rewards from different regularization methods is presented in Fig. 5, showing their performance throughout the training process on both experimental tasks. For the quadruped locomotion task, the results show that activation regularization, normalization, and the PPG method achieved similar performance. Despite this, their final cumulative rewards were statistically lower than those from the proposed discount regularization approach. The superiority of our method becomes even more pronounced in the challenging quadruped parkour task, where it consistently surpassed the other three baselines. We theorize that PPG's suboptimal performance might stem from its original design for image-based applications, which can introduce inductive biases poorly suited for low-dimensional observation spaces. Ultimately, this comprehensive comparison shows that the proposed method consistently secures the highest cumulative rewards across different tasks, highlighting its task-agnostic nature and its potential to enhance generalization in DRL.

TABLE III
PERFORMANCE OF DIFFERENT DISCOUNT REGULARIZATION FACTORS

Task \ γ_{reg}	0.94	0.96	0.98	0.99
Quadruped Locomotion	16.75±0.25	**18.05±0.05**	17.80±0.18	15.43±0.14
Quadruped Parkour	7.71±4.28	9.62±4.72	**12.46±0.15**	9.84±0.38
Total	24.46	27.67	**30.26**	25.27

Note: Bold values are indicate the best performance.

3) Ablation of Discount Regularization Factors: To justify the selection of the discount factor γ_{reg} for our proposed discount regularization, we conduct an ablation study on four distinct values: 0.94, 0.96, 0.98, and 0.99. The value $\gamma_{\text{reg}} = 0.98$ is adopted for our proposed method, while $\gamma_{\text{reg}} = 0.99$ corresponds to the baseline normal experiment detailed in Section IV-B2. To ensure a fair comparison, we train the robot with three different random seeds for each setting. After convergence, we evaluate performance by averaging the cumulative returns over the final 100 episodes for each run. We then report the mean and standard deviation across the three seeds. The results are summarized in Table III.

As shown in the table, for the quadruped locomotion task, all tested γ_{reg} values below 0.99 outperform the baseline. Specifically, $\gamma_{\text{reg}} = 0.96$ achieves the highest return, closely followed by $\gamma_{\text{reg}} = 0.98$. For the quadruped parkour task, however, only $\gamma_{\text{reg}} = 0.98$ yields a superior performance compared to the baseline. Furthermore, the aggregated return from both tasks is maximized when $\gamma_{\text{reg}} = 0.98$. Therefore, considering the overall performance across both environments, we select $\gamma_{\text{reg}} = 0.98$ for our discount regularization method.

4) Comparison of Training Efficiency: In this section, we select a specific morphology and train it using two distinct methods to evaluate and compare their training efficiency.

a) *Standard training*: Normative training under the specific morphology without pretraining.

b) *Fine-tuning*: The proposed method.

The comparison of training epochs is illustrated in Fig. 6, where (a) and (b) depict the results for different tasks. For the quadruped locomotion task, the cumulative reward begins to converge after 4000 epochs, while for the quadruped parkour task, convergence occurs after 2000 epochs. In contrast, when applying the fine-tuning method, both tasks achieve convergence as early as the 100th epoch. These findings validate that the proposed method can significantly accelerate the

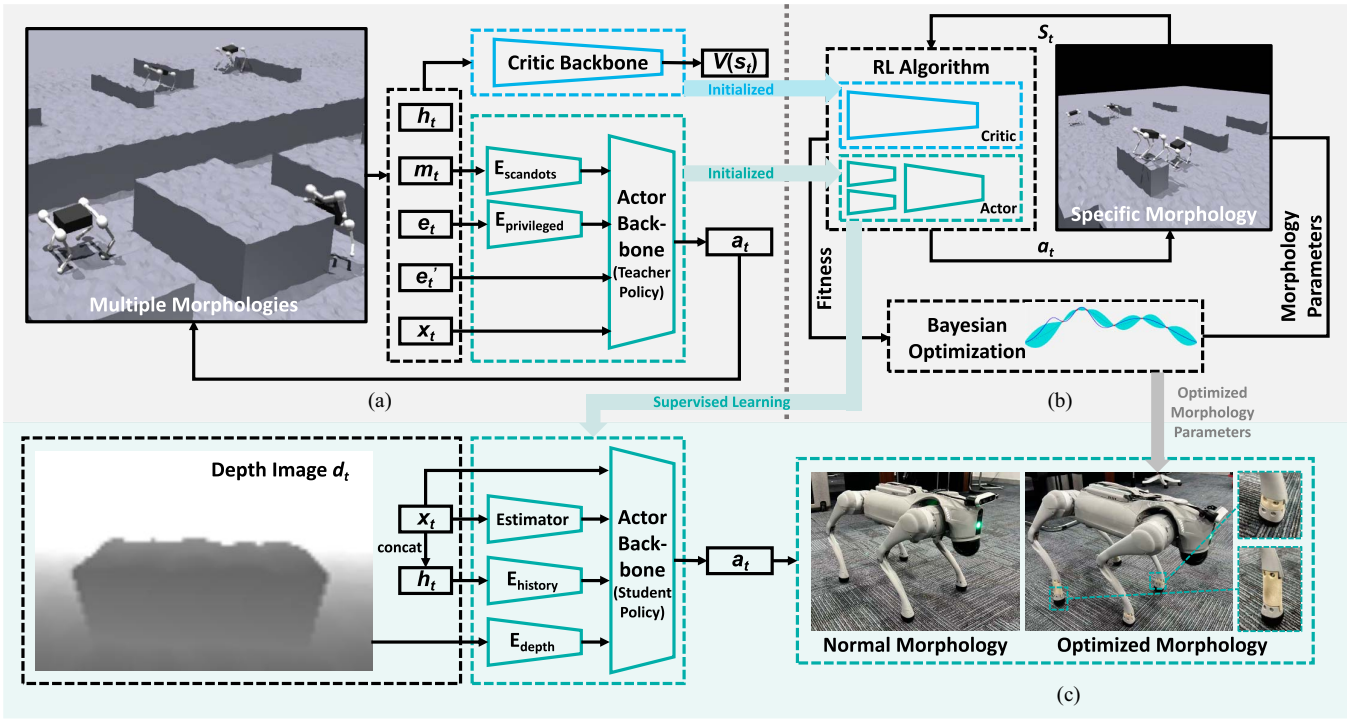


Fig. 3. Overview of the proposed codesign pipeline. The policy optimization utilizes the proximal policy optimization (PPO) algorithm [16] within an asymmetric actor–critic framework. Specifically, the privileged encoder maps e_t to a latent vector, while the scan-dots encoder transforms m_t into a terrain latent representation. The actor network receives these latent vectors combined with e_t' and x_t as inputs, whereas the critic network operates on the complete state information. (a) Pre-training. (b) Fine-tuning. (c) Deployment.

training process. For instance, when training 30 specific morphologies, standard training requires $30 \times 6000 = 1.8 \times 10^5$ epochs, whereas fine-tuning only needs $6000 + 30 \times 400 = 1.8 \times 10^4$ epochs. The number 6000 represents the pretraining steps, while 400 indicates the fine-tuning steps tailored for specific morphologies, equivalent to 6.67% of the former.

Furthermore, we conduct a comparative performance analysis of the two methods, as illustrated in Fig. 6. For the quadruped locomotion task, both methods converge at approximately 14, while for the quadruped parkour task, convergence occurs around 12. To quantitatively assess the differences, we compute the mean of the last 100 reward values for both methods. Specifically, for the first task, the reward discrepancy between the two methods is 1.19%; whereas for the second task, it is 2.08%. These results demonstrate that the proposed approach achieves performance comparable to standard training while maintaining a minimal margin of error, all while requiring only one-tenth of the training time.

5) Comparison of Codesign Baselines: In this section, we compare the proposed method with three codesign baselines. We first introduce these methods.

a) *Online-PCODP [9]*: This method uniformly samples design parameters within their range during training, treating them as privileged information. It assumes a policy conditioned on these parameters can achieve near-optimal performance.

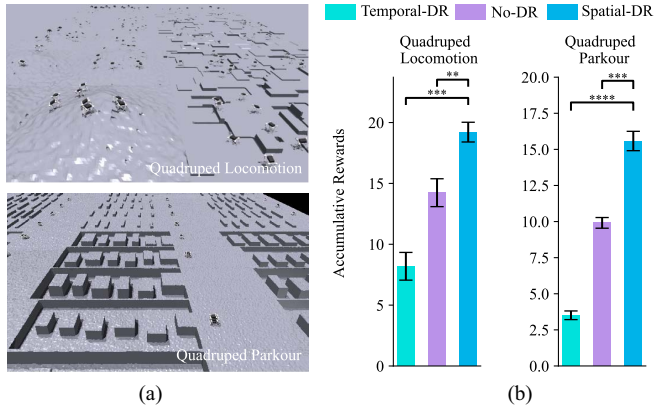


Fig. 4. Comparison of domain randomization methods. (a) Descriptions of various tasks. (b) Comparison of the cumulative reward values achieved by the robots under different domain randomization methods. Statistical significance was assessed using the T-test. $^*P < 0.05$; $^{**}P < 0.01$; $^{***}P < 0.001$; $^{****}P < 0.0001$.

b) *Offline-PCODP [6]*: This method employs online DRL to collect interaction data from many distinct morphologies and then trains a general policy via an offline DRL algorithm TD3-BC [28].

c) *EAT [8]*: Similar to Offline-PCODP, the primary difference lies in replacing the TD3-BC method with the decision transformer [29].

d) *Ours*: The proposed method.

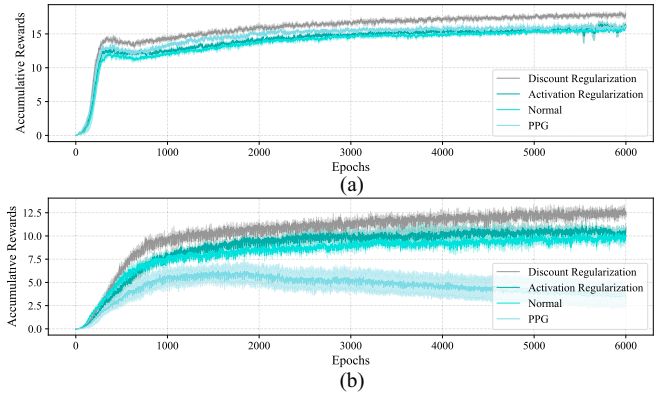


Fig. 5. Curve of cumulative rewards changes with the training steps. (a) Curves for the quadruped locomotion task. (b) Curves for the quadruped parkour task.

TABLE IV
RESULTS OF COMPARISON WITH CO-DESIGN BASELINES

Method \ Task	Quadruped Locomotion		Quadruped Parkour		Average Time Consumption
	Steps Walk	Hills Walk	Long Jump	High Jump	
Online-PCODP [9]	23.82	25.06	6.18	10.37	**4 h 46 min**
Offline-PCODP [6]	15.83	19.38	4.78	4.63	60 h 53 min
EAT [8]	3.02	0.95	0.66	0.71	60 h 29 min
Ours	**24.72**	**26.25**	**28.23**	**29.06**	7 h 5 min

Note: Bold values are indicate the best performance.

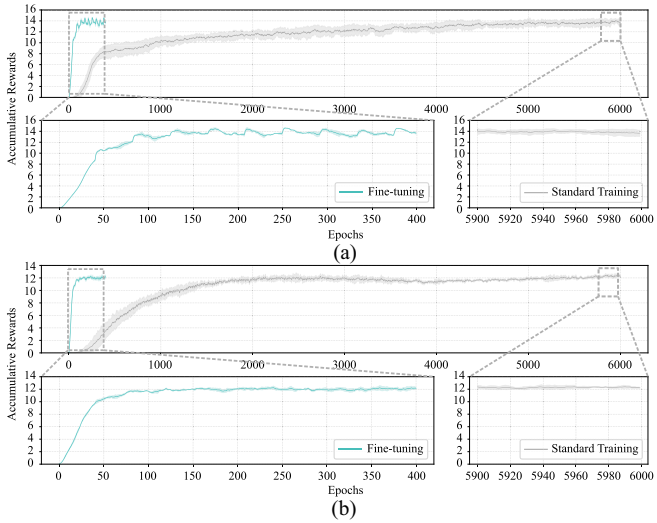


Fig. 6. Training efficiency comparison of fine-tuning and standard training. (a) Results for the quadruped locomotion task. (b) Results for the quadruped parkour task.

For quadruped locomotion and parkour tasks, we uniformly sampled 81 distinct morphologies from the defined parameter space. Fig. 7 visualizes cumulative rewards across methods with results averaged over three seeds. The proposed method consistently outperforms baselines across all morphologies in both tasks, as shown in Fig. 7(a) and (b)(iv), demonstrating strong generalization across morphologies and task complexities. Conversely, Fig. 7(a) and (b)(i) and 7(a) and (b)(ii) shows suboptimal rewards for both Online-PCODP and Offline-PCODP. Specifically, Online-PCODP achieves higher returns on later trained morphologies but degraded performance on earlier ones, indicating catastrophic forgetting, while Offline-PCODP performs well on its 16 training morphologies but fails to generalize to unseen ones. Both methods attain higher rewards on simpler tasks like locomotion than complex parkour tasks, indicating better generalization to low-complexity scenarios.

Finally, Fig. 7(a) and (b)(iii) shows EAT performing worst in both tasks. For this comparison, we implemented the EAT baseline [8] faithfully to its original architecture to evaluate its direct applicability to our more complex tasks. We hypothesize that this underperformance stems from its design targeting simple planar locomotion with low-dimensional state representations. Our tasks' complex terrain significantly increases the input state dimensionality, which appears to degrade the performance of the unmodified Transformer-based architecture. This result, obtained from what is a potentially suboptimal configuration for EAT in our setting, underscores the challenge of generalizing existing methods to more demanding environments without architectural adaptation.

Moreover, we selected two representative terrains from quadruped locomotion and quadruped parkour, respectively, and applied different methods to each of them. The fitness achieved by different methods is reported in Table IV. Notably, our proposed method attained the highest values across all four tasks, demonstrating the effectiveness of our pretraining-finetuning framework under the codesign paradigm.

Table IV also presents the training time consumed by each method. For Online-PCODP and our proposed approach, the reported time includes both the policy training and the time required for BO. In contrast, Offline-PCODP and EAT account for the time needed to train expert policies, collect expert demonstrations, train a generalization controller, and conduct BO. Although the proposed method incurs some additional time overhead due to the finetuning process embedded in BO, the overall training time remains significantly lower than that of Offline-PCODP and EAT, resulting in an 88% improvement in training efficiency. Considering both performance and time efficiency, the superiority of our method becomes evident.

C. Evaluation in Real World

1) Deployment Details: Real-world experiments are conducted on the Unitree Go2 EDU quadruped robot. The robot is equipped with a single inertial measurement unit (IMU) and 12 motor encoders that collectively provide critical state information, including body orientation, angular velocity, joint positions, and joint velocities. Depth images are captured at 10 Hz using the onboard Intel RealSense D435i camera. Image postprocessing and neural network inference are performed with LibTorch on the onboard NVIDIA Jetson Orin NX computing platform, while control actions are generated at 50 Hz. Communication between the robot's main control computer and the NX is implemented via ROS2.

The optimization objective in this task is given by (10) in Section III-C2. Additionally, owing to the impracticality of

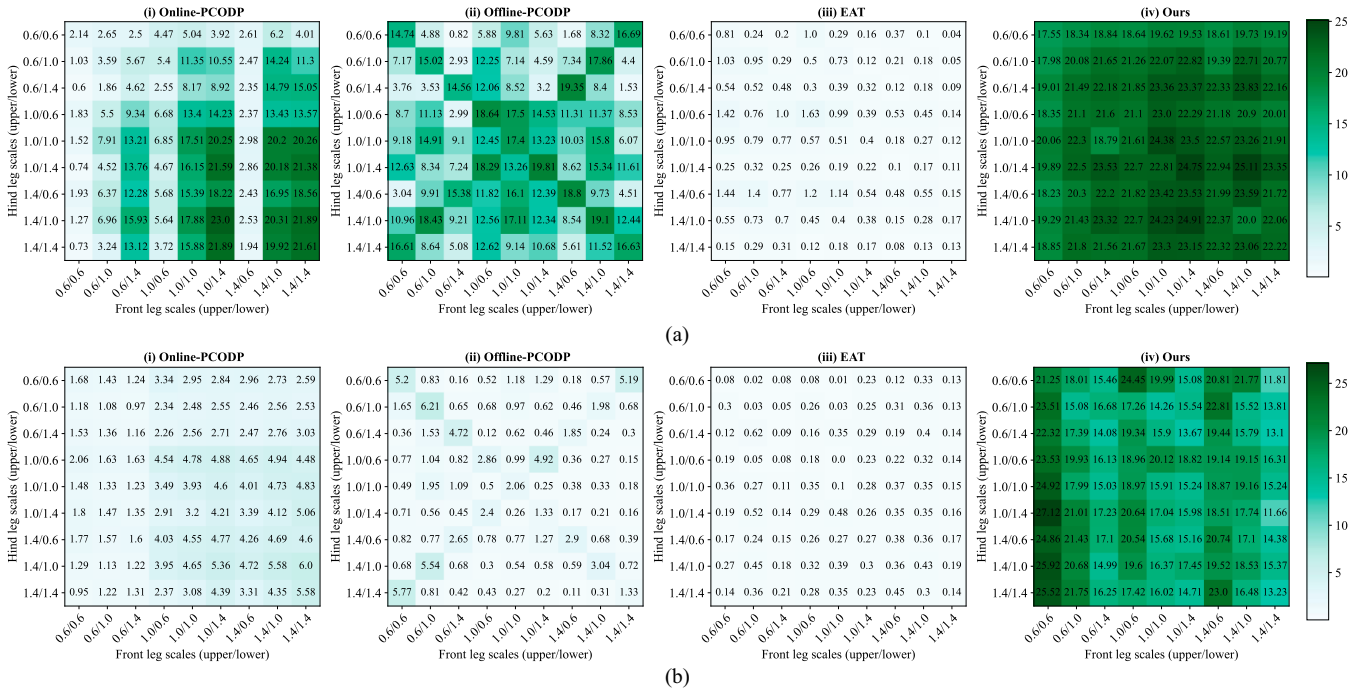


Fig. 7. Heatmap of codesign experiments. (a) Results for the quadruped locomotion task. (b) Results for the quadruped parkour task. Each block represents the cumulative rewards achieved using the corresponding control strategy under a given morphology. For Offline-PCODP and EAT, the 16 typical morphologies are located along the diagonal of the heatmap (excluding the center point).

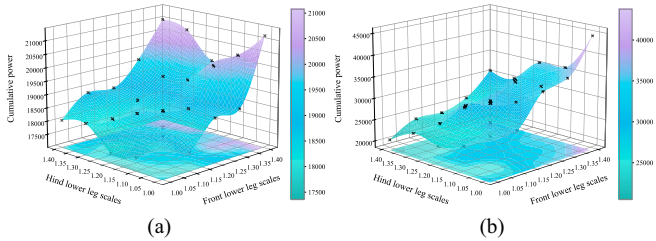


Fig. 8. Morphological fitness landscape analysis. (a) Results for the quadruped locomotion (slope traverse) task. (b) Results for the quadruped parkour (high jump) task.

modifying the robot's thigh structure, the optimization parameters are restricted to the lengths of the foreleg shanks and hindleg shanks, expressed as $\xi_{l,l=1,3} \in [1.0, 1.4]$. The structural component (as shown in Fig. 3) was fabricated from a glass-fiber-reinforced nylon composite to withstand the substantial external loads experienced during locomotion.

2) Morphological Fitness Landscape Analysis: To provide clear design insights, we constructed 5×5 fitness landscape maps over the design space based on the absolute value of (10), as shown in Fig. 8. The value in each block is the average fitness for a specific design, calculated over 250 episodes across 4096 environments.

The analysis reveals that the two tasks favor different morphologies. For the slope traverse task, the fitness landscape is relatively gentle, indicating that performance is less sensitive to morphological changes. The most effective configurations are clustered around the standard design or those with hindlimbs slightly longer than the forelimbs. Conversely, significantly extending the limbs, especially the forelimbs, generally increases energy consumption. This suggests that for stable locomotion, radical morphological changes are detrimental to energy efficiency. In contrast, the high jump task demonstrates a much stronger and clearer morphological preference. The results show that an asymmetric "short-forelimb, long-hindlimb" structure significantly reduces energy consumption. On the other hand, any "long-forelimb, short-hindlimb" configuration leads to a sharp increase in energy use and poor performance. This finding offers a clear design principle: different tasks require fundamentally different body structures. For highly dynamic actions like jumping, mimicking the asymmetric limb design of jumping animals is key to enhancing performance.

3) Physical Experiment Results: For the quadruped locomotion category, we use slope traverse as the representative task. As shown in Fig. 9(a), the energy consumption differences among the various morphologies on the slope are minimal, a finding that aligns with our preceding analysis. Moreover, the task's instantaneous power curve is smooth, lacking distinct peaks. As a result, the cumulative energy consumption increases almost linearly over time.

In contrast, the quadruped parkour category, represented by the high jump task, presents a starkly different energy curve. As shown in Fig. 9(b), this dynamic task is characterized by a prominent peak in instantaneous power, which leads to a sharp increase in cumulative energy consumption, clearly distinguishing it from slope traverse. This is precisely where the optimized "short-forelimb, long-hindlimb" asymmetrical

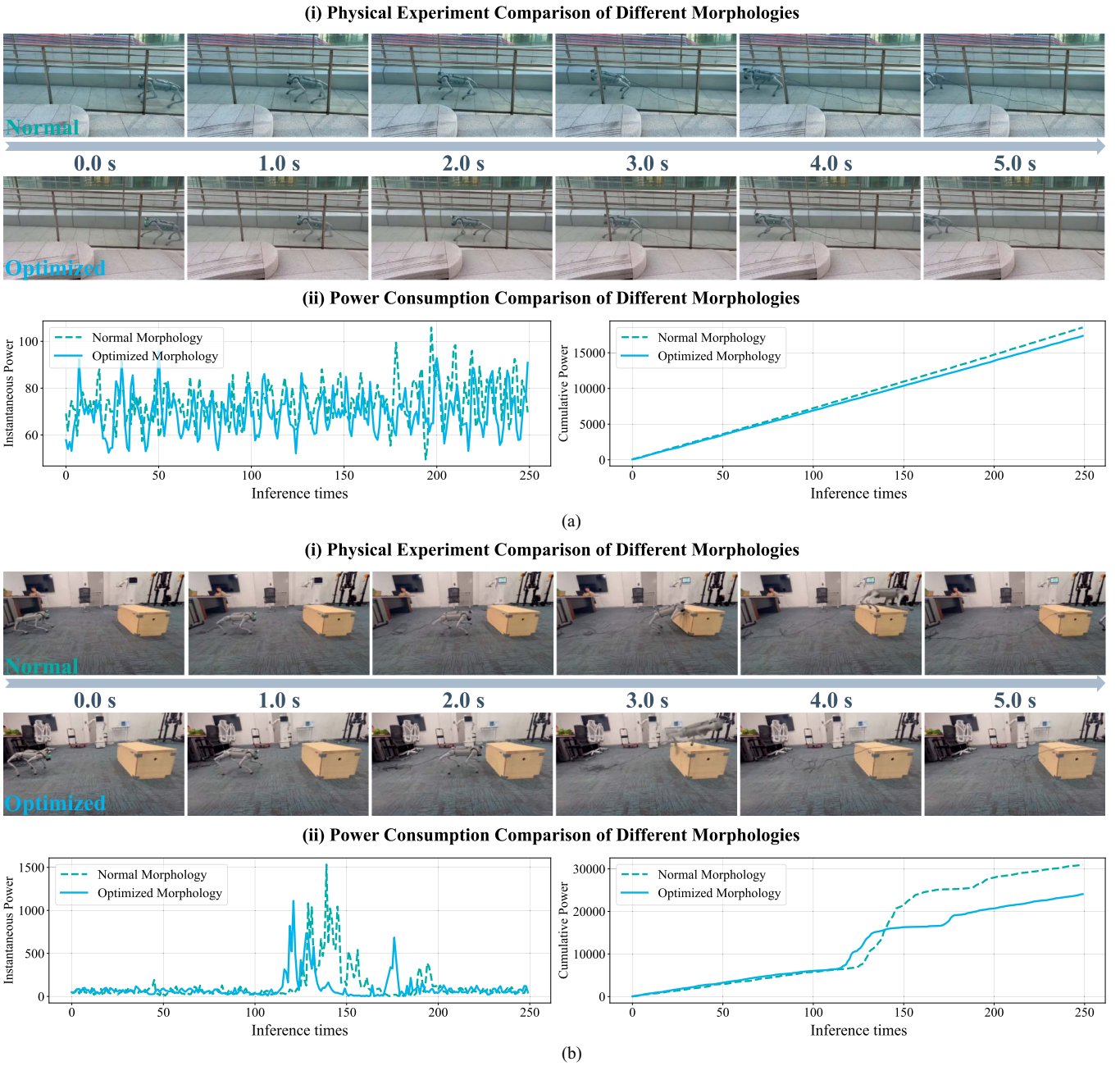


Fig. 9. Physical experiment results for different quadruped morphologies. (a) Locomotion task (slope traverse): (i) experimental snapshots; and (ii) comparison of instantaneous and cumulative power consumption. (b) Parkour task (high jump): (i) experimental snapshots; and (ii) comparison of instantaneous and cumulative power consumption. The task duration was 5 s, with the algorithm operating at 50 Hz, resulting in a total of 250 inference times.

morphology demonstrates its advantage. It significantly improves energy efficiency by lowering the peak power demand during the jump's ascent. The underlying mechanism is functional specialization: the longer hindlimbs serve as powerful levers for propulsion, while the shorter forelimbs enhance stability by minimizing the overturning moment. This design mirrors the limb specialization found in many jumping animals. However, it is important to note that since our optimization was constrained to the lower limbs, the resulting morphology should be considered a task-specific local optimum rather than a globally optimal design.

V. Conclusion

This article introduces a pretraining-finetuning framework for mechanical-control codesign in robotics. During the pretraining phase, we propose a methodology combining spatial domain randomization and discount factor regularization to train a generalizable model. The fine-tuning stage is embedded within the structural optimization loop, enabling rapid morphology-specific policy adaptation via the pretrained model during each iteration. This approach simultaneously ensures policy optimality across distinct morphologies and computational efficiency. By cooptimizing mechanical

structures and control policies, we significantly enhance quadruped robots' motion capabilities, expanding their deployment potential and operational efficacy in diverse environments.

However, we recognize several key limitations, which also point to directions for future research. A primary limitation is the constrained scope of the design space. While optimizing limb lengths produced significant performance gains, this set of parameters is not exhaustive. For instance, other important design factors such as joint configurations, mass distribution, and actuator dynamics are not considered, as investigating a broader design space was beyond the scope of this work. Second, the optimized morphology is highly task-specific. Our framework optimizes a robot's structure for a predefined objective, such as the high jump and slope traverse tasks from our experiments. Consequently, if the operational environment or task changes significantly, a new optimization process is required to find the new optimal design, which limits the robot's adaptability in real time.

To address these limitations, future work could focus on several promising directions. One valuable direction is to test the generalizability of our pretrained model on a richer and more diverse set of morphological parameters. Furthermore, a more forward-looking direction would be to explore the development of adaptive or reconfigurable robotic components. Such advancements would enable quadruped robots to dynamically adjust their structure in response to different tasks or terrains, enhancing their versatility and adaptability in diverse environments.

References

[1] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "DTC: Deep tracking control," *Sci. Robot.*, vol. 9, no. 86, 2024, Art. no. eadh5401.

[2] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "ANYmal parkour: Learning agile navigation for quadrupedal robots," *Sci. Robot.*, vol. 9, no. 88, 2024, Art. no. eadi7566.

[3] J. Qi, H. Gao, H. Su, M. Huo, H. Yu, and Z. Deng, "Reinforcement learning and sim-to-real transfer of reorientation and landing control for quadruped robots on asteroids," *IEEE Trans. Ind. Electron.*, vol. 71, no. 11, pp. 14392–14400, Nov. 2024.

[4] X. Liu, J. Wu, Y. Xue, C. Qi, G. Xin, and F. Gao, "Skill latent space based multigait learning for a legged robot," *IEEE Trans. Ind. Electron.*, vol. 72, no. 2, pp. 1743–1752, Feb. 2025.

[5] Á. Belmonte-Baeza, J. Lee, G. Valsecchi, and M. Hutter, "Meta reinforcement learning for optimal design of legged robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 12134–12141, Oct. 2022.

[6] C. Chen, P. Xiang, H. Lu, Y. Wang, and R. Xiong, "C 2: Co-design of robots via concurrent-network coupling online and offline reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE, 2023, pp. 7487–7494.

[7] C. Chen, P. Xiang, J. Zhang, R. Xiong, Y. Wang, and H. Lu, "Deep reinforcement learning based co-optimization of morphology and gait for small-scale legged robot," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 4, pp. 2697–2708, Aug. 2024.

[8] C. Yu, W. Zhang, H. Lai, Z. Tian, L. Kneip, and J. Wang, "Multi-embodiment legged robot control as a sequence modeling problem," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE, 2023, pp. 7250–7257.

[9] F. Bjelonic et al., "Learning-based design and control for quadrupedal robots with parallel-elastic actuators," *IEEE Robot. Automat. Lett.*, vol. 8, no. 3, pp. 1611–1618, Mar. 2023.

[10] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Joint optimization of robot design and motion parameters using the implicit function theorem." in *Proc. Robot.: Sci. Syst.*, vol. 8, 2017, pp. 10–15607.

[11] G. Fadini, T. Flayols, A. Del Prete, N. Mansard, and P. Souères, "Computational design of energy-efficient legged robots: Optimizing for size and actuators," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE, 2021, pp. 9898–9904.

[12] T. Wang, Y. Zhou, S. Fidler, and J. Ba, "Neural graph evolution: Towards efficient automatic robot design," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[13] A. Gupta, S. Savarese, S. Ganguli, and L. Fei-Fei, "Embodied intelligence via learning and evolution," *Nat. Commun.*, vol. 12, no. 1, 2021, Art. no. 5721.

[14] C. Schaff, D. Yunis, A. Chakrabarti, and M. R. Walter, "Jointly learning to construct and control agents using deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE, 2019, pp. 9798–9805.

[15] F. De Vincenti, D. Kang, and S. Coros, "Control-aware design optimization for bio-inspired quadruped robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE, 2021, pp. 1354–1361.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[17] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024, pp. 11443–11450.

[18] J. Tan, et al., "Sim-to-real: Learning agile locomotion for quadruped robots," 2018, *arXiv:1804.10332*.

[19] V. Makoviychuk, et al., "Isaac gym: High performance GPU based physics simulation for robot learning," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round 2)*, 2021.

[20] S. Moon, J. Lee, and H. O. Song, "Rethinking value function learning for generalization in reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 34846–34858.

[21] K. W. Cobbe, J. Hilton, O. Klimov, and J. Schulman, "Phasic policy gradient," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 2020–2027.

[22] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.

[23] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Proc. Conf. Robot Learn. (CoRL)*, 2022, pp. 138–149.

[24] Q. Wu, Z. Fu, X. Cheng, X. Wang, and C. Finn, "Helpful DoggyBot: Open-world object fetching using legged robots and vision-language models," 2024, *arXiv:2410.00231*.

[25] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. Conf. Robot Learn.*, PMLR, 2022, pp. 91–100.

[26] Z. Luo et al., "MorAL: Learning morphologically adaptive locomotion controller for quadrupedal robots on challenging terrains," *IEEE Robot. Automat. Lett.*, vol. 9, no. 5, pp. 4019–4026, May 2024.

[27] G. Feng, et al., "GenLoco: Generalized locomotion controllers for quadrupedal robots," in *Proc. Conf. Robot Learn.*, PMLR, 2023, pp. 1893–1903.

[28] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20132–20145.

[29] L. Chen et al., "Decision transformer: Reinforcement learning via sequence modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15084–15097.

Ci Chen received the B.S. degree in agricultural mechanization and automation from the Northeast Agricultural University, Harbin, China, in 2018, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2024.

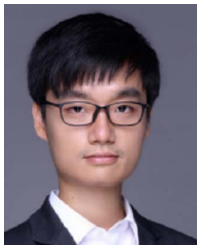
She is currently a Postdoctoral Researcher with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China. Her research interests include deep reinforcement learning and legged robots.

Yifei Yu is currently working toward the B.S. degree in automation engineering with Shanghai University of Electric Power, Shanghai, China, in 2023.

His research interests include mechanism design and robotics.

Haoqi Han received the B.S. degree in mechatronics engineering from Beijing Institute of Technology, Beijing, China, in 2018. He is currently working toward the Ph.D. degree in computer science and engineering with Shanghai Jiao Tong University, Shanghai, China.

His research interest includes biomimetic mechanisms.

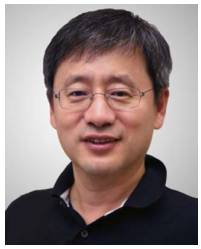


Yue Wang (Member, IEEE) received the Ph.D. degree in control science and engineering from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2016.

He is currently a Professor with the Department of Control Science and Engineering, Zhejiang University. His research interests include mobile robotics and robot perception.

Rong Xiong (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2009.

She is currently a Professor with the Department of Control Science and Engineering, Zhejiang University. Her research interests include motion planning and Simultaneous Localization and Mapping.



Hesheng Wang (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from Harbin Institute of Technology, Harbin, China, in 2002, and the M.Phil. and Ph.D. degrees in automation and computer-aided engineering from The Chinese University of Hong Kong, Hong Kong, China, in 2004 and 2007, respectively.

He is currently a Distinguished Professor with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China. His research interests include visual servoing, intelligent robotics, computer vision, and autonomous driving.

Dr. Wang is an Associate Editor of *Robotic Intelligence and Automation* and *International Journal of Humanoid Robotics*, a Senior Editor of IEEE/ASME TRANSACTIONS ON MECHATRONICS, and the Editor-in-chief of *Robot Learning*. He served as an Associate Editor of IEEE TRANSACTIONS ON ROBOTICS from 2015 to 2019 and IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING from 2021 to 2023. He was the General Chair of IEEE/RSJ IROS 2025, IEEE ROBIO 2022, and IEEE RCAR 2016.