Informed Repurposing of Quadruped Legs for New Tasks

Fuchen Chen[1] and Daniel M. Aukes[1]

***Abstract*— Redesigning and remanufacturing robots are infeasible for resource-constrained environments like space or undersea. This work thus studies how to evaluate and repurpose existing, complementary, quadruped legs for new tasks. We implement this approach on 15 robot designs generated from combining six pre-selected leg designs. The performance maps for force-based locomotion tasks like pulling, pushing, and carrying objects are constructed via a learned policy that works across all designs and adapts to the limits of each. Performance predictions agree well with real-world validation results. The robot can locomote at 0.5 body lengths per second while exerting a force that is almost 60% of its weight.**

I. Introduction

In resource-constrained environments the opportunity for redesigning or remanufacturing robots can be limited. Certain kinds of exploration enabled by robots require pre-manufacturing parts in advance and bringing them with you. This can often be seen in the redundant systems employed in space or undersea missions, where the cost of transporting replacement parts is not feasible.

Motivated by these scenarios, we aim to study how to evaluate combinations of existing, complementary, but not completely redundant parts and select the one that best suits the desired job. As shown in Fig. 1, our approach has been applied to a quadruped robot whose legs can be fabricated in various configurations based on a common template, and their behaviors can be tuned across a wide range of geometric and physical parameters. This work also focuses on force-based locomotion tasks, requiring the robot to perform positive work upon external objects in situations unanticipated during the original design step. This type of repurposing may become necessary and useful for our targeted applications and is the focus of our paper.

This work is based on our recently submitted work that formulates a pipeline for the design, fabrication, learning, evaluation, and selection of laminate legs with embedded compliance [1]. In that work, we looked at the role that combining the design process with a learned locomotion policy can have on selecting legs for achieving various velocity-based locomotion metrics. Our current work uses a limited set of leg designs suggested by it.

The contributions of this work are summarized as follows: 1) an extension of the design, fabrication, learning, evaluation, and selection pipeline to include force-based locomotion tasks and asymmetric robot designs, improving the generality of our approach; 2) a demonstration of constructing performance maps for a limited set of leg designs to repurpose them for different tasks and study roles of asymmetry in quadruped's force capabilities; 3) and insights from real-world validations on the capabilities of the simulation environments for centimeter-scaled compliant robots under loads.

[1]Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, Arizona, 85212, USA. Email: {fchen65, danaukes}@asu.edu

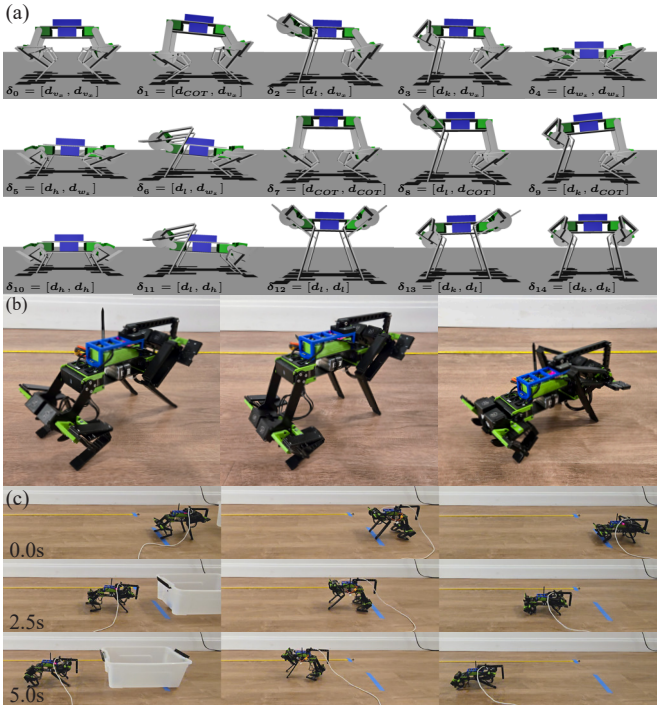

Fig. 1. (a) 15 robot designs are generated from mixing and matching 6 pre-selected leg designs; reinforcement learning is used to evaluate their performance for force-based locomotion tasks. (b) Three of the designs are tested with the learned policy in the real world, performing tasks such as (c) pulling a box, carrying a payload, and walking freely. More details of the experiments are available in the supplementary video.

The rest of the paper is organized in the following way: Section I-A provides some more background literature related to this work; Section II-A mainly recaps the leg and robot designs; Section II-B explains the learning framework for teaching our robot to perform force-based locomotion tasks; Section II-C listed the experiments performed; Section III discusses the evaluation results of the various robot designs; and this paper concludes in Section IV.

A. Background

For legged systems, the spring-loaded inverted pendulum (SLIP) model is a simple but effective model for describing the fundamental dynamics of walking and running [2], [3]. It is also widely used in robotics to study the design of robotic legs [4], [5], [6], [7], [8]. Most of the existing works on leg designs usually concern only pure locomotion tasks without

doing work to external objects. For example, the design of iSprawl hexapod involves tuning the leg compliance with the SLIP model to achieve smooth running gaits [4]. As for Salto, mechanical advantage tuning can also be paired with the series elasticity in the SLIP model to achieve extreme vertical jumping agility [7]. One of our earlier works studies the effect of nonlinear series leg stiffness for jumping [8].

However, as the robots and tasks become complex with more contact and friction behaviors with the floor and surrounding objects, this simple model will become less accurate. More recent works have started to employ model-based or learning-based approaches to add more capabilities beyond pure locomotion to legged robots [9], [10], [11]. An adaptive manipulation controller paired with a model predictive controller is proposed for regulating the contact force to push an unknown object on unknown terrain [9]. Reinforcement learning can also be leveraged to train policies for quadrupeds to push and pull objects with their whole body [11] or an attached arm [10].

While there are many advancements in the control of legged robots for loco-manipulation tasks, few works treat the robot design as a variable and study its impact on performance. Interestingly, some simulation-based works deal with co-designing or evolving robots or creatures for both locomotion [12], [13] and manipulation tasks such as carrying, pushing, and throwing objects [14], [15]. In contrast, this and our previous work focus on bridging the simulation and the real world with accessible laminate design and fabrication techniques and extensive real-world validations.

Since our approach is grounded in reality, some artificial and empirical decisions are necessary to keep the design space compact. Allowing asymmetry for our robot – that is, having different front and rear legs – is inspired by existing works showing some benefits of asymmetric designs: an asymmetric segmented body results in smaller cost of transport (COT) [16] and asymmetric body-mass distribution leads to lower stride frequency and smaller duty factor [17]. Our concept of combining limited parts for different performances and tasks also takes inspiration from the design optimization and control of modular robots [18], [19], though we emphasize passive components in a distinctly different form factor.

II. Methods

A. Robot Design

1) Robot: As shown in Fig. 1, the robot used in this work has a footprint similar in area to an adult's hand and weighs under 500 g. Each of its legs is actuated by two identical servo motors acting in series. The abduction and adduction joints have been omitted. The robot has a battery, an IMU sensor, and a microcontroller on board. The policy is run on a laptop and communicates with the robot through a USB cable. All four legs can be swapped with different designs, as discussed next. More details of the robot can be found in previous work [1].

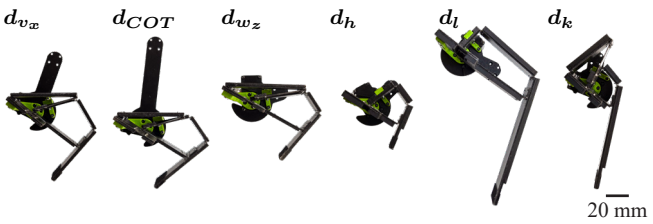

Fig. 2. All 6 leg designs used to form different robot designs. Details of the design and fabrication of the tunable leg can be found in our previous work [1].

TABLE I
THEORETICAL LEG PROPERTIES

	d_{v_x}	d_{w_z}	d_{COT}	d_h	d_l	d_k
Total Leg length (m)	0.12	0.06	0.14	0.06	0.14	0.12
Foot Travel length (m)	0.04	0.02	0.02	0.02	0.10	0.02
Moment arm (m)	0.067	0.067	0.067	0.033	0.067	0.022
Parallel Stiffness (N/m)	22	22	45	0	0	202
Series Stiffness (N/m)	360	360	360	1800	270	4050

2) Legs: Inspired by the SLIP model, the leg used in our current work is designed to function similarly to a pogo stick. It is composed of four main components: a femur link connecting the hip and knee joint, a four-bar linkage that converts knee rotation into linear foot translation going through the hip, a parallel spring that resists leg retraction, and a series spring that allows the foot to move even if the knee servo is fixed. The tunable properties of the leg include the maximum leg length from the foot to the hip, the total travel length of the foot, the moment arm or transmission ratio for converting knee rotation to foot travel, and the parallel and series leg stiffness along the foot travel direction. The leg in the current study aligns with our previous work [1].

As shown in Fig. 2, six leg variants have been selected and fabricated for this work. These previously-identified designs represent the diverse design space of the tunable leg and possess unique geometric and dynamic characteristics including fast longitudinal running d_{v_x}, fast turning d_{w_z}, low cost of transport d_{COT}, short height d_h, long foot travel d_l, and high parallel and series stiffness d_k. Their theoretical leg properties are listed in Table I.

3) Mix and Match: Allowing asymmetry between the front and rear legs is a simple way to expand the robot design space with limited leg selections. In theory, the six leg variants can generate 21 robot designs based on the combinations with repetition formula [20]. However, since some combinations with large leg length differences resulted in failures to walk during preliminary testing, they were excluded. Fifteen robot designs $\delta_{0,1,\ldots,14}$ are left, as shown in Fig. 1 (a).

B. Learning Force-based Locomotion

1) Policy Design: This work uses reinforcement learning to train a neural network policy to control the robot's locomotion under external forces in simulation. A learning-based approach was selected principally because the complex

design and control space made a first-principles approach less feasible. A single policy was trained for use across all robot designs. We assume that this saved time by learning similar fundamental behaviors, and reduced the chance of the policy prematurely settling at local solutions optimal for one design. The proposed policy has the following inputs and outputs, which is an adaptation of our previous work:

$$\pi(\boldsymbol{a_t}|\boldsymbol{v_t^{cmd}}, \boldsymbol{p}, \boldsymbol{f}, \boldsymbol{q_t}, \boldsymbol{\dot{q}_t}, \boldsymbol{g_t}, \boldsymbol{w_t}, \boldsymbol{a_{t-1}}, \boldsymbol{d_f}, \boldsymbol{d_r}) \,. \quad (1)$$

As shown in Fig. 3, the main goal of the policy is to make the robot follow a velocity command $\boldsymbol{v_t^{cmd}} = [v_x^{cmd}, w_z^{cmd}]$ consisting of the longitudinal speed and turning speed while an external force $\boldsymbol{f} = [f_x, f_z]$ is applied to the robot at $\boldsymbol{p} = [p_x, p_z]$. The available action $\boldsymbol{a_t} \in \boldsymbol{R}^8$ for the policy represents the desired joint positions sent to the servos. Since our robot lacks active abduction and adduction, we do not command lateral speeds v_y^{cmd}; force commands therefore lie in the sagittal plane. The velocity and force are defined in a reference frame whose x axis is parallel to the floor, xz plane aligns with the robot's heading, and origin coincides with the body's center, since the robot's body is not necessarily level across all leg pairs.

By specifying the external force vector and its application point about different locations on the robot, it is possible to simulate a variety of loading conditions similar to real-world scenarios. A force applied to the front of the robot toward the body indicates that the robot is pushing a load. A vertical downward force applied to the robot body represents carrying payloads. These combinations also naturally encode moments applied to the robot, which can significantly impact load-carrying ability. This formulation ignores inertia related to external loads since we focus on the robot's performance during steady-state locomotion.

The policy has access to a range of sensory feedback that is available on the actual prototype, including joint angles $\boldsymbol{q_t} \in \boldsymbol{R}^8$, joint velocities $\boldsymbol{\dot{q}_t} \in \boldsymbol{R}^8$, body orientation $\boldsymbol{g_t} \in \boldsymbol{R}^3$ expressed as the gravity direction in the robot's body frame, and body angular velocities $\boldsymbol{w_t} \in \boldsymbol{R}^3$. The policy also observes its last action $\boldsymbol{a_{t-1}}$. Additionally, since the policy must accommodate different front and rear leg designs, the design vector $\boldsymbol{d_f}$ and $\boldsymbol{d_r}$ are provided in the observations, containing the values of the tunable leg properties mentioned in Section II-A.2. We also expose applied force values to the policy. The real-world experiments are thus designed such that the forces applied to the robot are known and remain relatively constant throughout the test. One final point of note: since we are more interested in the robot's capability of doing positive work such as pulling or pushing loads, the direction of the longitudinal speed v_x^{cmd} remains opposite to the direction of the horizontal force f_x during all evaluations.

The policy is a fully connected neural network with three hidden layers of size $[512, 256, 128]$ and ELU activations operating at 100 Hz. The actions and observations are scaled and offset to approximately within the range of $[-1, 1]$ before being fed into the policy.

2) Curriculum Design: A curriculum for the training process is deployed for two main reasons: it can be challenging for a robot to learn how to move under large external forces initially, so a gradual increase in the external force is desirable; and the bounds of the external force that the robot can handle are also unknown, requiring a way to automatically identify them during training.

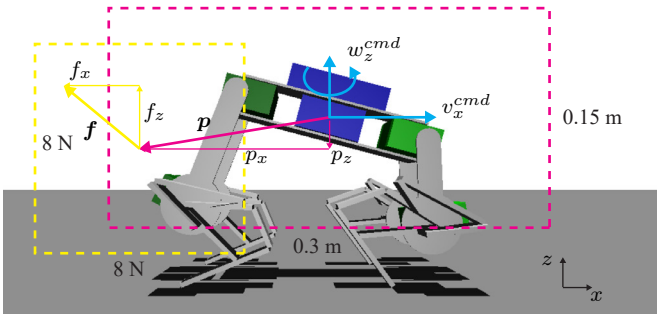

Fig. 3. Definitions of the velocity command, external force application point, and external force vector for the policy. The bounding boxes for the application point and force vector are also shown.

For a single robot design, we construct a discretized force application space $\hat{\boldsymbol{\Phi}} \in \boldsymbol{R}^4$ containing force applications $\hat{\boldsymbol{\phi}} = [\hat{\boldsymbol{p}}, \hat{\boldsymbol{f}}] = [\hat{p}_x, \hat{p}_z, \hat{f}_x, \hat{f}_z]$ with a step size $\boldsymbol{s_{\hat{\Phi}}} = [0.1\,m, 0.05\,m, 1\,N, 1\,N]$. To prevent the curriculum from exerting forces that may damage the actual physical robot and limit the force application space $\hat{\boldsymbol{\Phi}}$ to a reasonable size so that the training does not take too long, the lower and upper bounds for $\hat{\boldsymbol{\phi}}$ are $\pm[0.1\,m, 0.05\,m, 3.5\,N, 3.5\,N]$. These essentially limit $\hat{\boldsymbol{\Phi}}$ to a $3 \times 3 \times 8 \times 8$ grid, which has a high enough resolution to reveal trends while remaining tractable. This curriculum strategy is a modification to the one used in our previous work that only employs two dimensions – in that case the longitudinal and turning speed.

At the beginning of the training, the four points near the origin $\hat{\boldsymbol{\phi}} = [0, 0, \pm 0.5, \pm 0.5]$ are in the space $\hat{\boldsymbol{\Phi}}$. As the training progresses, the space is expanded by adding neighbors of a force application that the robot can handle, meaning that $ema(\bar{r}_v) > r_v^{th}$ where $ema(\bar{r}_v)$ is the exponential moving average (EMA), with a 0.1 smoothing factor, of the average velocity reward of an episode. The velocity reward is defined as $r_v = 2exp(-25\|\boldsymbol{v} - \boldsymbol{v^{cmd}}\|^2)$ and the threshold is $r_v^{th} = 1.21$. The neighbors of a force application $\hat{\boldsymbol{\phi}}$ is defined as $\hat{\boldsymbol{\phi}}_{\boldsymbol{n}} = \hat{\boldsymbol{\phi}} + \boldsymbol{s_{\hat{\Phi}}} \cdot \boldsymbol{I_{row}}$ where $\boldsymbol{I_{row}}$ is one of the rows of the identity matrices $\boldsymbol{I_4}$ and $-\boldsymbol{I_4}$, which is an extension to the 2 dimensional 4-connectivity. For each episode, the actual force applied to the robot is $\boldsymbol{\phi}$ which is uniformly chosen within $(\hat{\boldsymbol{\phi}} - 0.5\boldsymbol{s}_{\hat{\boldsymbol{\phi}}}, \hat{\boldsymbol{\phi}} + 0.5\boldsymbol{s}_{\hat{\boldsymbol{\phi}}})$, where $\hat{\boldsymbol{\phi}}$ is also uniformly sampled from the latest $\hat{\boldsymbol{\Phi}}$. Since different robot designs can have different learning difficulties and limits, the training keeps a curriculum for each leg design and updates them individually based on their respective policy performance.

In addition, although the achievable velocities of the robot may also depend on applied forces, we do not include them in the curriculum to keep training time manageable. The bounds for $\boldsymbol{v^{cmd}}$ are set to $\pm[0.3\,m/s, 0.3\,rad/s]$, a relatively small range for our robot. For each force application, the velocity command is uniformly sampled from this range with the constraint that it is opposite to the horizontal force f_x.

3) Learning Environment: For each episode during training, the robot is equipped with a design that is uniformly sampled from the 15 designs; it tries to track the velocity commanded under an external force. The external force is converted manually to a wrench applied to the CoM of the robot body. A random disturbance wrench with magnitudes up to $\pm 10\%$ of the applied one is also added. Domain randomization including servo dynamics parameters, floor properties, sensor noise, and system latency is also applied; details can be found in our previous work.

Modifications to the previous reward terms are also necessary for successful learning: the hip joint is no longer required to move around the center point, but the difference between the average left and right hip joint angles over a second needs to be penalized to generate more symmetric gaits; the body pitch does not need to be level to account for the difference between front and rear legs; and vertical body speeds are also penalized to stay away from a suboptimal bouncing gait.

C. Experiments

1) Training and Evaluation: We trained the policy using Proximal Policy Optimization for 300k updates over about 110 hours, after which the growth rate of the curriculum space slowed. Details of the software implementation, hyperparameters, and hardware can also be found in [1].

To evaluate the performance of the learned policy and the capability of each robot design, the policy was tasked to control the robot in the same randomized simulation environments subjected to every force application $\hat{\phi}$ added to the space $\hat{\mathbf{\Phi}}$ during the training. Each trial lasted for 2 seconds, and the average velocity reward r_v and COT starting from one second was recorded to get the steady-state performance. $COT = \frac{\bar{p}}{mg|\bar{v}_x|}$ where $\bar{p}$ and $\bar{v}_x$ are the average positive mechanical power and average actual longitudinal speed. For a single $\hat{\phi}$, 200 trials were performed and the results were averaged. The COT was only considered if the commanded longitudinal speed was greater than 0.1 m/s to avoid getting large values that masked actual trends.

2) Real-world Validations: To validate the performance of the policy and reveal trends discovered from the evaluation results, we tested three robot prototypes. Details of the design selection are discussed in the following section. For each design, the policy was tasked to walk for 5 seconds at a constant velocity $\boldsymbol{v_t^{cmd}} = [\pm 0.2\,m/s, 0\,rad/s]$ under different loading conditions made up from a horizontal force $f_x \in [0\,N, 1\,N, 2\,N, 3\,N]$ and a vertical force $f_z \in [0\,N, -2\,N]$ applied to the robot's body center $\boldsymbol{p} = [0\,m, 0\,m]$. As shown in Fig. 4, to apply horizontal forces, a box of various weights was attached to the back of the robot with a string; the robot therefore pulled the weight of the box in all tests. The string was roughly level to the floor; it was aligned to go through the robot body's center. The weights placed in the box were tuned to ensure that the kinetic friction roughly matched the desired values. Vertical forces were added by strapping weights to the robot's main body. Off-center loading conditions were not tested since implementing them was less feasible and repeatable given available resources.

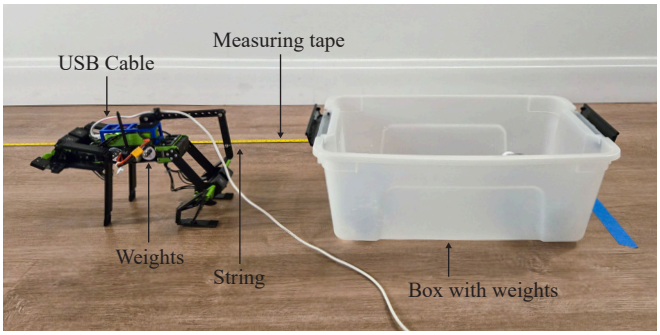

Fig. 4. Experimental setup for the real-world validations. The robot pulls a box of weights through a string. The speed is measured with the measuring tape and on-board sensory data are collected from the USB cable.

For each condition, the average speed was obtained by measuring the distance between a start and end point with a measuring tape and dividing it by the elapsed time. The mechanical power was estimated from the on-board joint velocity and current data. The COT was calculated similarly to the simulation. 3 trials were conducted for every condition.

III. Results and Discussions

A. Policy Evaluation

While the evaluation was performed in simulated environments with various randomized parameters, the results should still exhibit general trends across different designs and be useful for guiding the design selection for different tasks. The evaluation results are visualized in Fig. 6 and 7. Zoom-in views of the velocity reward results for the robot design $\boldsymbol{\delta_0}$ are shown in Fig. 5. As in Fig. 5 (a), the 3×3 grid represents the different force application points around the robot. For example, the top left cell as in Fig. 5 (b) depicts the average performance of the robot subjecting to forces applied to the region above the rear legs in the sagittal plane, which means $\hat{\boldsymbol{p}} = [-0.01, 0.05]$ and $p_x \in (-0.015, -0.05)$ and $p_z \in (0.025, 0.075)$. Within a cell, a single pixel represents the average performance for a specific force range. For example, a pixel below and to the left of the origin indicated by a black dot means $\hat{\boldsymbol{f}} = [-0.5, -0.5]$ and $f_x \in (-1, 0)$ and $f_z \in (-1, 0)$. Since the longitudinal speed command is always opposing the horizontal force direction. Within each cell, the left half separated by the dashed line implies that the robot is moving forward or to the right in Fig. 1 (a) and the right half is for moving backward. Although grouped together due to their shared leg pairs, they actually represent two very different scenarios and use cases.

1) Individual Designs: When looking at individual designs, the cells in the middle column have a shape similar to an isosceles triangle, which should be related to the friction cone concept: a larger normal force results in larger frictional forces, meaning that the robot can exert a larger pulling or pushing force on its load. The ratio of the maximum horizontal force to the vertical one is related to the friction coefficient. This ratio approximately falls within the range from 3/7=0.43 to 4/7=0.57, which is expected considering

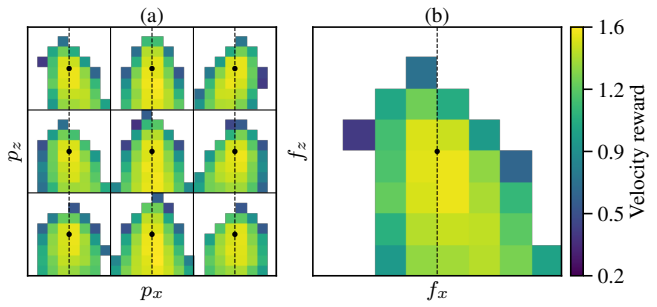

Fig. 5. Evaluation results of the velocity reward for the robot design $\boldsymbol{\delta_0}$. (a) shows the complete grid and (b) is a zoom-in view of the top left cell.

that the friction coefficient range for the simulated environments was from 0.4 to 1.0, and the evaluation results were averaged over many trials under different conditions.

Another trend is that the area of a cell generally becomes smaller as the force application points move away from the body center horizontally. This is related to the additional induced moment decreasing the normal forces for the feet. This trend is less obvious in the vertical direction since the step size for the vertical offset is half of the horizontal one.

These findings demonstrate that our training method is capable of estimating the limits for the robot's capability subjected to simulated physics. Moreover, it is also able to identify gaits that work within these limits at the same time. The benefits of this approach can reduce the time and effort required for manually designing and tuning controllers across different robot designs, thus permitting more time for additional design iteration and validation.

2) Across Designs: Comparing the performance across robot designs, differences are visible for both the velocity tracking and COT metrics. For each design, four metrics are further established: the total velocity reward for moving across forward and backward locomotion, and the average COT across forward and backward cases as well. Distinguishing the direction of motion is justified since many designs with different front and rear legs may be more or less suitable for different use cases. Their performance scores are plotted in Fig. 8, where the values below the diagonal are for the robot moving forward.

Visible in this plot are distinct performance differences between moving forward and backward. Design δ_{11} is the most obvious, with a 17% velocity reward difference and a 33% average COT difference. However, since even symmetric designs also display some asymmetric performance, part of the difference can be attributed to the unbalanced capability of the learned policy and the influence of the stochasticity of the tests on the overall results. For the total velocity reward, the average difference of the asymmetric designs is 6%, slightly larger than the 4% of the symmetric ones. Therefore, it is safe to conclude that the heading direction does not significantly affect the pulling and pushing capabilities of our robot. This is also reasonable, considering that the legs function similarly to a pogo stick, which should not have many directional preferences on relatively smooth floors, as in this work. If the floors had more variations and asperities,

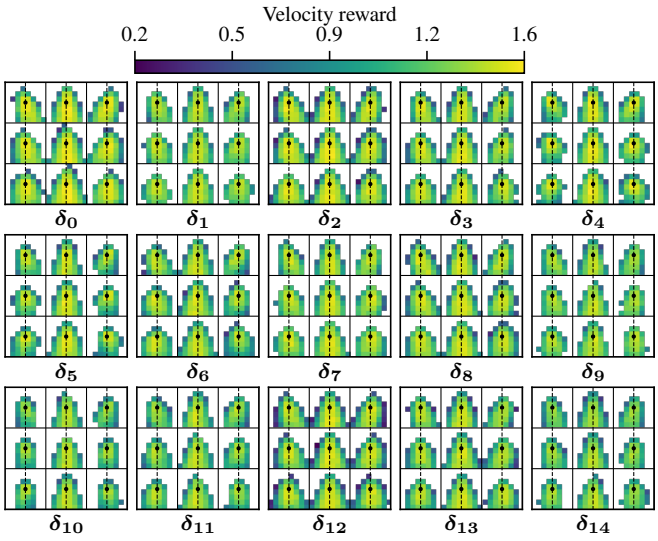

Fig. 6. Evaluation results of the velocity reward for all robot designs. A lighter color indicates better performance.

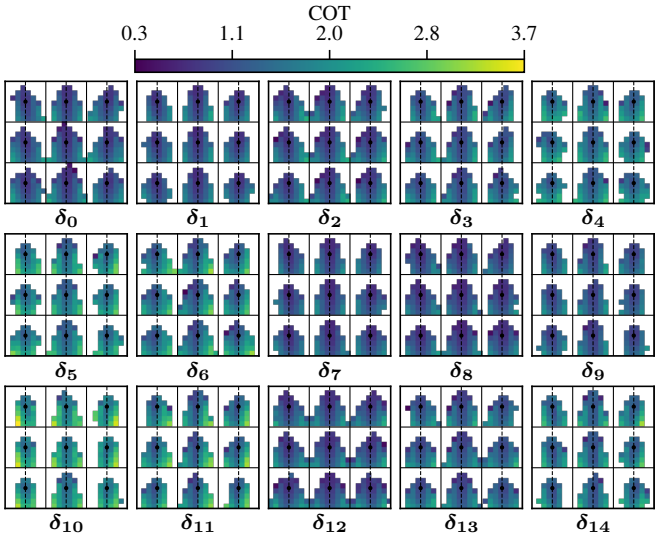

Fig. 7. Evaluation results of the COT for all robot designs. A darker color indicates better performance.

the different angles of attack of the legs resulting from leg length differences could lead to more interesting behavior.

Fig. 8 is also useful for comparing the performance of robots with different leg combinations. For total velocity reward, robots with legs $\boldsymbol{d_{v_x}}$ and $\boldsymbol{d_l}$ outperform the others. We hypothesize that their longer foot travel allows them to do more work within each gait cycle. Studies involving more leg designs are needed to verify this. For COT, robots with the leg $\boldsymbol{d_{COT}}$ perform better than their peers. Our previous study had also selected $\boldsymbol{d_{COT}}$ for its low COT during pure locomotion. Biological studies have shown that longer legs generally have lower COT [21], which agrees with our previous results. This work further demonstrates that this trend applies to robots in loaded conditions.

There is no conclusive evidence that combining different leg designs can achieve much better performance than their symmetric counterparts. For example, $\boldsymbol{d_l}$ paired with itself

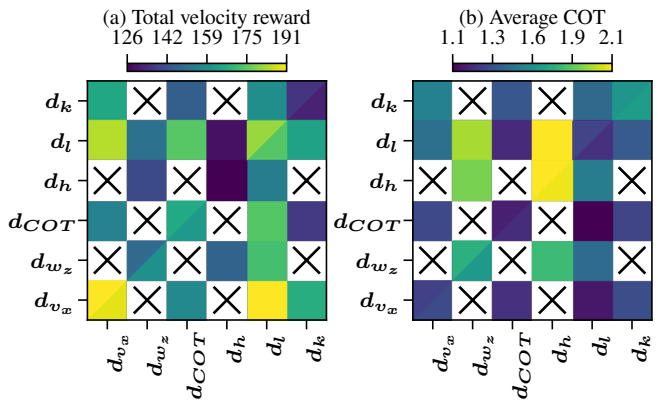

Fig. 8. Total velocity reward and Average COT for different combinations of legs moving forward (below the diagonal) and backward. The X's in specific squares indicate leg pairs that were not included during training.

TABLE II

PERFORMANCE OF SELECTED DESIGNS FOR REAL-WORLD VALIDATIONS

	Robot Designs	$\boldsymbol{\delta_2}$	$\boldsymbol{\delta_8}$	$\boldsymbol{\delta_{11}}$
	Legs	$[d_l, d_{v_x}]$	$[d_l, d_{COT}]$	$[d_l, d_h]$
Sim	Total velocity reward	**375**	**348**	283
	Average COT	1.29	**1.13**	1.79
Real	Average speed (m/s)	**0.10**	**0.11**	0.08
	Average COT	1.79	**1.20**	4.61

or d_{v_x} perform almost equally well in terms of the total velocity reward, but the performance degrades if it is paired with other worse leg designs. d_{COT} has similar stories for the average COT. Interestingly, this could indicate that the performance of asymmetric robots can be roughly predicted by averaging its symmetric versions, which could greatly reduce the computation time because only symmetric designs need to be learned and evaluated.

B. Real-world Validations

Three robot designs were selected for real-world validations as listed in Table II. All of them have leg d_l as the rear legs so that we can verify that changing the leg combinations does alter the performance. Moreover, they all have some unique performance characteristics based on the evaluation in the simulation environments. $\boldsymbol{\delta_0}$ and $\boldsymbol{\delta_8}$ are better at withstanding external loads than $\boldsymbol{\delta_{11}}$ since their total velocity reward is higher. $\boldsymbol{\delta_8}$ has the lowest COT.

As marked in Table II, the real-world results agree well with the simulation in terms of performance rankings, demonstrating that the proposed pipeline is a useful tool for selecting designs based on task objectives. To be noted, the average speed of all the conditions for each design is used instead of total velocity reward because velocity data over time are not available.

The detailed data for the real-world experiments are plotted in Fig. 9 and 10. As the magnitude of the pulling force increases, the actual speed of the robot drops and the COT increases. The COT is also higher when the robot is loaded. The speed error compared to the commanded speed of 0.2 m/s is relatively large for several potential reasons:

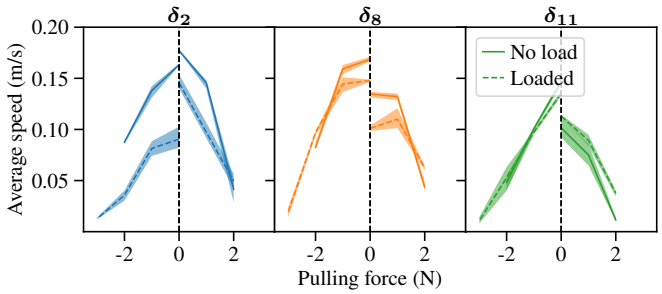

Fig. 9. Average speed of the selected robot designs under various loading conditions. The shaded regions indicate the range of the data.

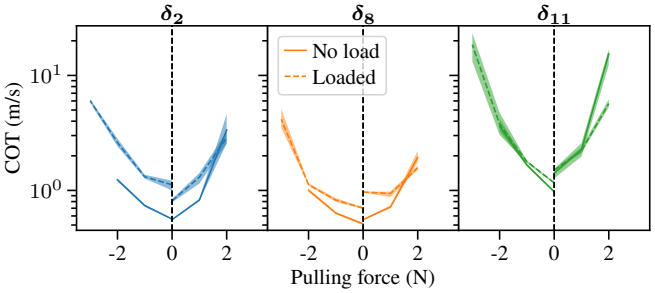

Fig. 10. COT of the selected robot designs under various loading conditions. The shaded regions indicate the range of the data.

there are noticeable unintended and unmodeled compliance and backlash for both legs and servos, making the locomotion less stable and more shaky, especially under load; the traction provided by the friction tape wrapped around the narrow feet is less reliable; and the additional inertia of the load is not considered during training in the simulation.

IV. CONCLUSIONS

Given quadruped designs generated from a set of preselected legs, this work proposes a pipeline for evaluating their performance for force-based locomotion tasks that require doing positive work upon external objects such as pushing, pulling, and carrying loads. Our learning-based approach produces a policy that works across designs and adapts to the limits of each. Evaluation results enable us to select designs suitable for different performance goals, and the real-world validations show good agreement. We show that asymmetric robot designs produce more performance variety, but do not necessarily lead to significant improvement.

We believe that the proposed approach based on reinforcement learning is a powerful tool for constructing performance maps of a complex robot design space in which dynamics are discontinuous, nonlinear, and stochastic. This work showcases the potential for producing customized robots for different applications, and discovering underlying design principles that may often be hidden by system complexity. In the future, we plan to improve the computational efficiency of our framework and extend it beyond legged robots.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 1944789.

References

[1] F. Chen and D. M. Aukes, "Curating Tunable, Compliant Legs for Specialized Tasks," *International Journal of Robotics Research*, accepted, 2025.

[2] R. M. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, "A Simply Stabilized Running Model," *SIAM Journal on Applied Dynamical Systems*, vol. 2, no. 2, pp. 187–218, Jan. 2003.

[3] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, Aug. 2006, publisher: Royal Society.

[4] S. Kim, J. E. Clark, and M. R. Cutkosky, "iSprawl: Design and Tuning for High-speed Autonomous Open-loop Running," *The International Journal of Robotics Research*, vol. 25, no. 9, pp. 903–912, Sept. 2006.

[5] K. C. Galloway, J. E. Clark, and D. E. Koditschek, "Design of a Multi-Directional Variable Stiffness Leg for Dynamic Running," in *Volume 10: Mechanics of Solids and Structures, Parts A and B*. Seattle, Washington, USA: ASMEDC, Jan. 2007, pp. 73–80.

[6] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 932–950, July 2013.

[7] D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," *Science Robotics*, vol. 1, no. 1, p. eaag2048, Dec. 2016.

[8] F. Chen and D. M. Aukes, "Direct Encoding of Tunable Stiffness Into an Origami-Inspired Jumping Robot Leg," *Journal of Mechanisms and Robotics*, vol. 16, no. 031012, Mar. 2023.

[9] M. Sombolestan and Q. Nguyen, "Hierarchical Adaptive Loco-manipulation Control for Quadruped Robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 12 156–12 162.

[10] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, "Learning Whole-Body Manipulation for Quadrupedal Robot," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 699–706, Jan. 2024, conference Name: IEEE Robotics and Automation Letters.

[11] T. Portela, G. B. Margolis, Y. Ji, and P. Agrawal, "Learning Force Control for Legged Manipulation," May 2024, arXiv:2405.01402 [cs, eess].

[12] N. Cheney, R. MacCurdy, J. Clune, and H. Lipson, "Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding," *ACM SIGEVOlution*, vol. 7, no. 1, pp. 11–23, Aug. 2014.

[13] 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 Transactions on Graphics*, vol. 39, no. 6, pp. 1–16, Dec. 2020.

[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, publisher: Nature Publishing Group.

[15] J. S. Bhatia, H. Jackson, Y. Tian, J. Xu, and W. Matusik, "Evolution Gym: A Large-Scale Benchmark for Evolving Soft Robots," 2022, publisher: arXiv Version Number: 1.

[16] L. T. Phan, Y. H. Lee, D. Y. Kim, H. Lee, and H. R. Choi, "Hybrid quadruped bounding with a passive compliant spine and asymmetric segmented body," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 3387–3392, iSSN: 2153-0866.

[17] H. Zou and J. Schmiedeler, "The effect of asymmetrical body-mass distribution on the stability and dynamics of quadruped bounding," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 711–723, Aug. 2006, conference Name: IEEE Transactions on Robotics.

[18] J. Hu, J. Whitman, M. Travers, and H. Choset, "Modular Robot Design Optimization with Generative Adversarial Networks," in *2022 International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, May 2022, pp. 4282–4288.

[19] J. Whitman, M. Travers, and H. Choset, "Learning Modular Robot Control Policies," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 4095–4113, Oct. 2023.

[20] R. A. Brualdi, *Introductory Combinatorics*, 5th ed. Upper Saddle River, N.J: Pearson, Jan. 2009.

[21] H. Pontzer, "Effective limb length and the scaling of locomotor cost in terrestrial animals," *Journal of Experimental Biology*, vol. 210, no. 10, pp. 1752–1761, May 2007.