2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

Reinforcement learning for freeform robot design

Muhan Li, David Matthews, Sam Kriegman
Northwestern University

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610048

***Abstract*—Inspired by the necessity of morphological adaptation in animals, a growing body of work has attempted to expand robot training to encompass physical aspects of a robot's design. However, reinforcement learning methods capable of optimizing the 3D morphology of a robot have been restricted to reorienting or resizing the limbs of a predetermined and static topological genus. Here we show policy gradients for designing freeform robots with arbitrary external and internal structure. This is achieved through actions that deposit or remove bundles of atomic building blocks to form higher-level nonparametric macrostructures such as appendages, organs and cavities. Although results are provided for open loop control only, we discuss how this method could be adapted for closed loop control and sim2real transfer to physical machines in future.**

I. Introduction

Usually, control policies are optimized on a robot or embodied agent with a hand designed physical form. However, it has been shown that a robot's physical structure can facilitate or obstruct policy optimization [2, 4, 7, 11]. This suggests that it may be useful to expand search to encompass the physical parameters of an agent as well, searching for structures that increasingly ease policy training.

Inspired by the evolution of animals, the automatic design of robots has been primarily achieved using evolutionary algorithms [4, 7, 11, 12, 17, 21, 30]. Although formal equivalences have been shown between learning and evolution [36], evolutionary robotics relies entirely on random phylogenetic "actions" (mutations) to modify the robot's design, without any bias toward favorable outcomes (beneficial mutations). That is, mutations at the beginning and end of evolution followed the same distribution and were not conditional on the state of the robot (its phenotype). Evolutionary and learning algorithms differ also in the kinds of problems they have successfully solved; and, until now, no reinforcement learning algorithm has been shown to be capable of freeform robot design.

Reinforcement learning (RL) and RL-adjacent methods have been used to lengthen or truncate a segmented torso and its pairs of jointed legs [39]; resize a quadruped's four legs [8, 9, 27] or a hexapod's six [32]; dynamically reorganize six modular limbs during behavior [26]; add or remove limbs at eight predefined locations [28]; and extrude four limbs from a torso that branch as unbalanced binary trees of depth three [37]. But, in every case, the basic geometric shape and internal structure of the robot's torso and limbs were predetermined and could not change during optimization. Also, these algorithms could not alter the robot's topology (number of voids).

Voids and pores could be useful for robots as they allow for internal carrying of objects, increase the robot's strength to weight ratio, and add surfaces for catalysis and heat exchange. Our approach uses thousands of atomic elements—voxels—as building blocks of macrostructures, which allows the number, placement, and 3D shape of limbs and voids to be optimized, simultaneously, with minimal assumptions.

Spherical particles [3] and cubic cells [19] have been used in previous studies to automatically optimize robot geometry and topology in a nonparametric manner, but they have yet to be leveraged for this purpose by RL methods. This gap in the literature is due in part to the *sui generis* nature of the design problem. Freely adding, reshaping and removing body parts is fundamentally different than resizing limbs and re-weighting synapses. The search landscape is much less forgiving of missteps: Adding or removing even a single voxel along the underside of a bipedal walker's foot, for instance, could have catastrophic behavioral consequences. Historically, this relegated the problem to selection among random variations. But recent years have witnessed a sea change in robot design.

Wang et al. [35] used RL to design of 2D agents composed of 49 elastic voxels. However, it was unclear if the agent's geometry varied during training, and if it did, if revisions were applied non-randomly. When optimizing the agent for locomotion, the agent's elasticity and motor layout were revised, but its square body shape remained unchanged.

Ma et al. [22] also took a voxel-based approach. Instead of RL, gradient information from differentiable simulation was used to interpolate between human-selected basis shapes. To ensure numerical stability, these body shapes were required to share same topology. The learner was thus trapped in the design space between hand-designed shapes and could not alter the robot's topology. Also, the robots in [22] were suspended in a liquid without contact modeling, which can be challenging to implement in a differentiable fashion, and without which land-based behaviors cannot be realized. This is an important distinction for AI as land affords cognitive opportunities to agents that are absent in aquatic environments (e.g. long-range vision and planning [24]).

Matthews et al. [23] used differentiable simulation for freeform design and optimization of terrestrial yet 2D robots. The two dimensional design space was extruded into a third dimension yielding a physical 3D body plan. More recently, others [5, 38] have extended differentiable design to three dimensions. While the robot from [23] was built and found to retain its behavior, it is unclear how methods that rely on differentiable physics can proceed without a custom built simulation for every new task the robot faces. RL, in contrast, requires only high-level reward definitions and can thus design physical machines directly without recourse to simulations.

Thus, we here introduce a policy-gradient method for de-

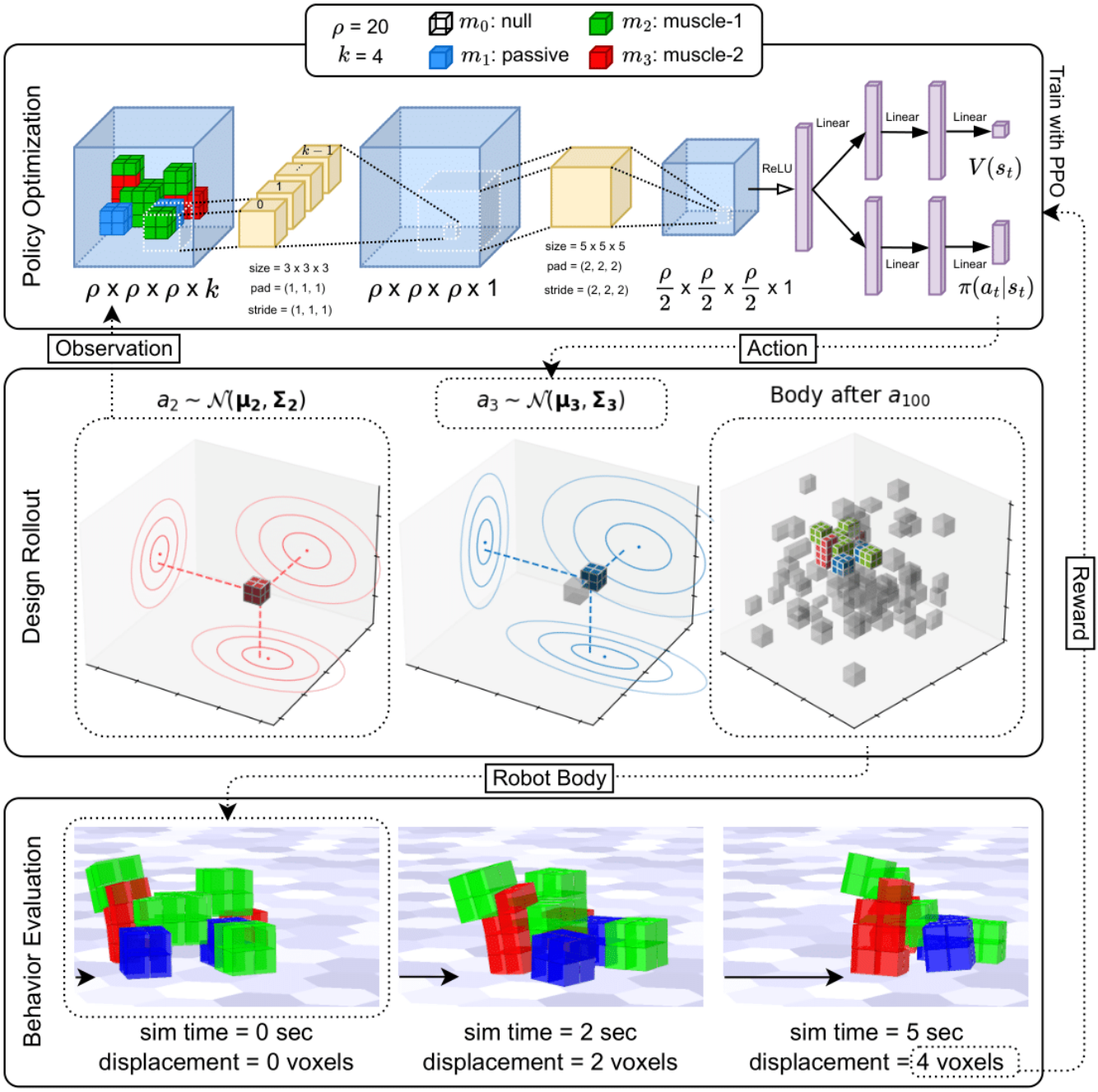

Fig. 1. **Freeform robot design.** A policy (*top row*) was trained to design increasingly motile robots, and predict their locomotive ability (critic; $V(s)$), using a sequence of 100 design actions that freely position, overwrite or remove bundles of muscles (green and red) and passive tissue (dark blue) to form a robot (*middle*) whose behavior in simulation (*bottom*) determines the policy's reward (youtu.be/ybaEVDGvkTE).

signing robots with freeform morphology. The resulting robots are freeform in the sense that they can be generated with any 3D shape and any internal 3D organization of materials and voids, at any given cartesian resolution.

II. Methods

A. The environment.

Robots are here generated along a $\rho \times \rho \times \rho$ voxel grid $\mathbf{G}$, yielding bodies with morphological resolution ρ. We here set $\rho = 20$. Each voxel within a robot's body consists of a central point mass and up to six Euler-Bernoulli beams connecting to its neighbors (if any) on each face. This allows for elastic twisting and stretching of one voxel relative to another. Actuation is implemented by increasing and decreasing the rest length of a voxel's beams, which produces volumetric expansion and contraction. A Coulomb friction model is used for the surface plane. For more details about the simulated physical environment, see [13].

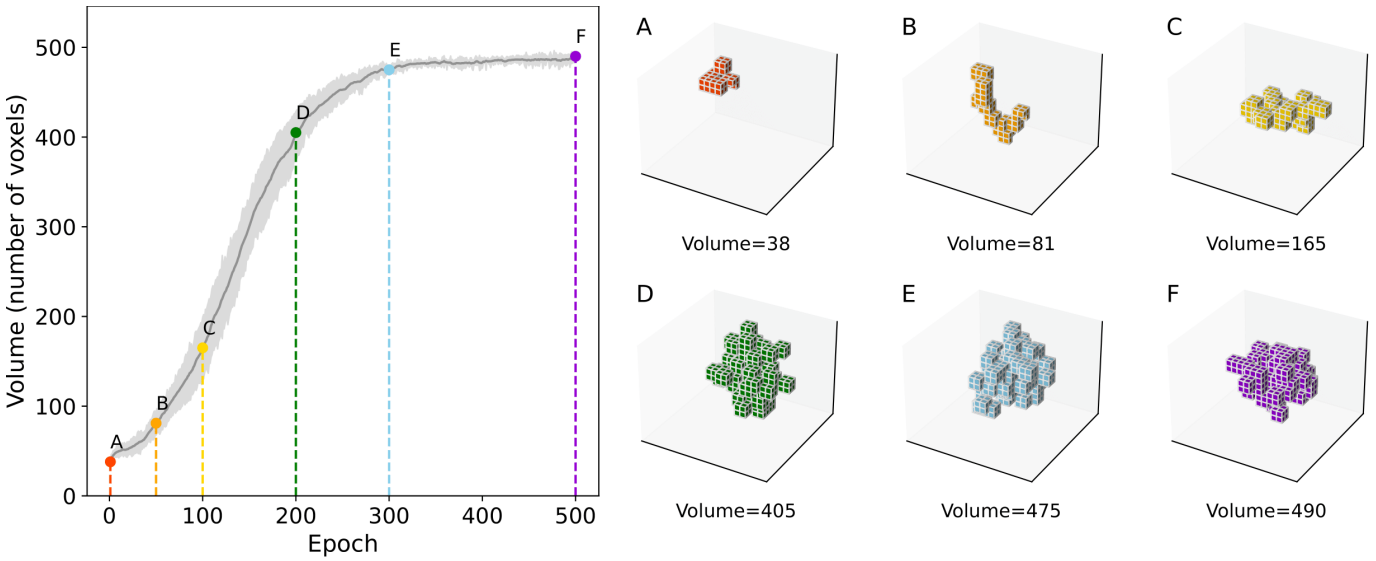

Fig. 2. **Learning to design large nonparametric bodies.** *Left:* Mean reward (volume; dark gray curve) and its 99% Normal confidence interval (light gray bands) across 5 independent learning trials. *Right:* six bodies (**A-F**) sampled along the reward curve from least to most voluminous.

B. The state space.

The state space $s_t \in \{1,0\}^{\rho\times\rho\times\rho\times k}$ describes the robot's body plan: the 3D position $\mathbf{x}$ of each voxel in $\mathbf{G}$ and k channels for 1-hot encoded material properties $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$, where m_0 is null material (empty space). If $\mathbf{m}$ is the zero vector then the null material is selected. We here set $k = 4$ with m_1 defined as passive tissue, and m_2, m_3 defined as muscles that actuate in anti-phase at 4 Hz with amplitude $\pm 10\%$. Materials m_1, m_2, m_3 all have Young's modulus 10^5 Pa (the stiffness of silicone rubber), density 1500 kg/m^3, Possion's ratio of 0.35, and coefficients of 1 and 0.5 for static and dynamic friction, respectively.

C. The action space.

Each sampled action $\tilde{a}_t = (\mathbf{x}, \mathbf{m})$ deposits a bundle of voxels b_t centered at $\mathbf{x}$, with material properties $\mathbf{m}$. We here set the bundle to be a $2\times2\times2$ cube of voxels. Actions follow a multivariate diagonal gaussian distribution $a_t \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with seven independent dimensions corresponding to the 3D position and the 4D material properties of the bundle. The support of the action distribution $a_t \in \mathbb{R}^7$ is mapped to that of $\mathbf{G}$ with clipping $f(a) = 1/2 + \text{clip}(a, -2, 2)/4$, which ensures that all bundles are centered inside of $\mathbf{G}$, but may "hang over" the edge. If voxels of two or more bundles overlap, the material properties of the most recent action take priority.

D. The reward.

After a sequence of T design actions, $\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_T$, the largest contiguous collection of voxels is taken to be the robot's body, $\mathcal{B}$, which is then evaluated in a physics-based virtual environment [13] and assigned a behavioral reward equal to the net displacement of the robot. The behavioral reward is only assigned to the reward of last step r_T and zero is assigned to rewards of other steps $r_1, r_2, \ldots, r_{T-1}$. Since rewarding for locomotion in terms of net displacement can create a local optimum of ever taller morphologies that utilize falling as the main method of displacement to reach a higher reward, we included a burn-in period in which the robot is given time to settle under gravity prior to recording the robot's initial location.

E. The policy.

The policy (Fig. 1) was trained using PPO [29] and comprises an actor $\pi(a_t|s_t)$, which takes design actions as described above in Sect. II-C, and a design critic $V(s_t)$, which is trained to approximate the accumulated discounted reward at design step t:

$$V(s_t) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t \right], \tag{1}$$

with discount factor $\gamma = 0.99$. The actor and critic share a base convolutional network but have independent fully connected layers, since parameter sharing [33] has been shown to simplify training and speed convergence. The base convolution network outputs the hidden embedding h_t of state s_t. The policy distribution and the value function is thus defined as:

$$h_t = \text{CNN}(s_t) \tag{2}$$

$$\boldsymbol{\mu} = \text{MLP}_{\pi,\boldsymbol{\mu}}(h_t) \tag{3}$$

$$\boldsymbol{\Sigma} = e^{\text{MLP}_{\pi,\boldsymbol{\Sigma}}(h_t)} \tag{4}$$

$$\pi(a_t|s_t) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{5}$$

$$V(s_t) = \text{MLP}_V(h_t) \tag{6}$$

The two MLPs: $\text{MLP}_{\pi,\boldsymbol{\mu}}$ and $\text{MLP}_{\pi,\boldsymbol{\Sigma}}(h_t)$ in Eq. 3 and 4 were implemented as a single MLP_π. The Xavier initialization method was used [6] to ensure that the untrained policy

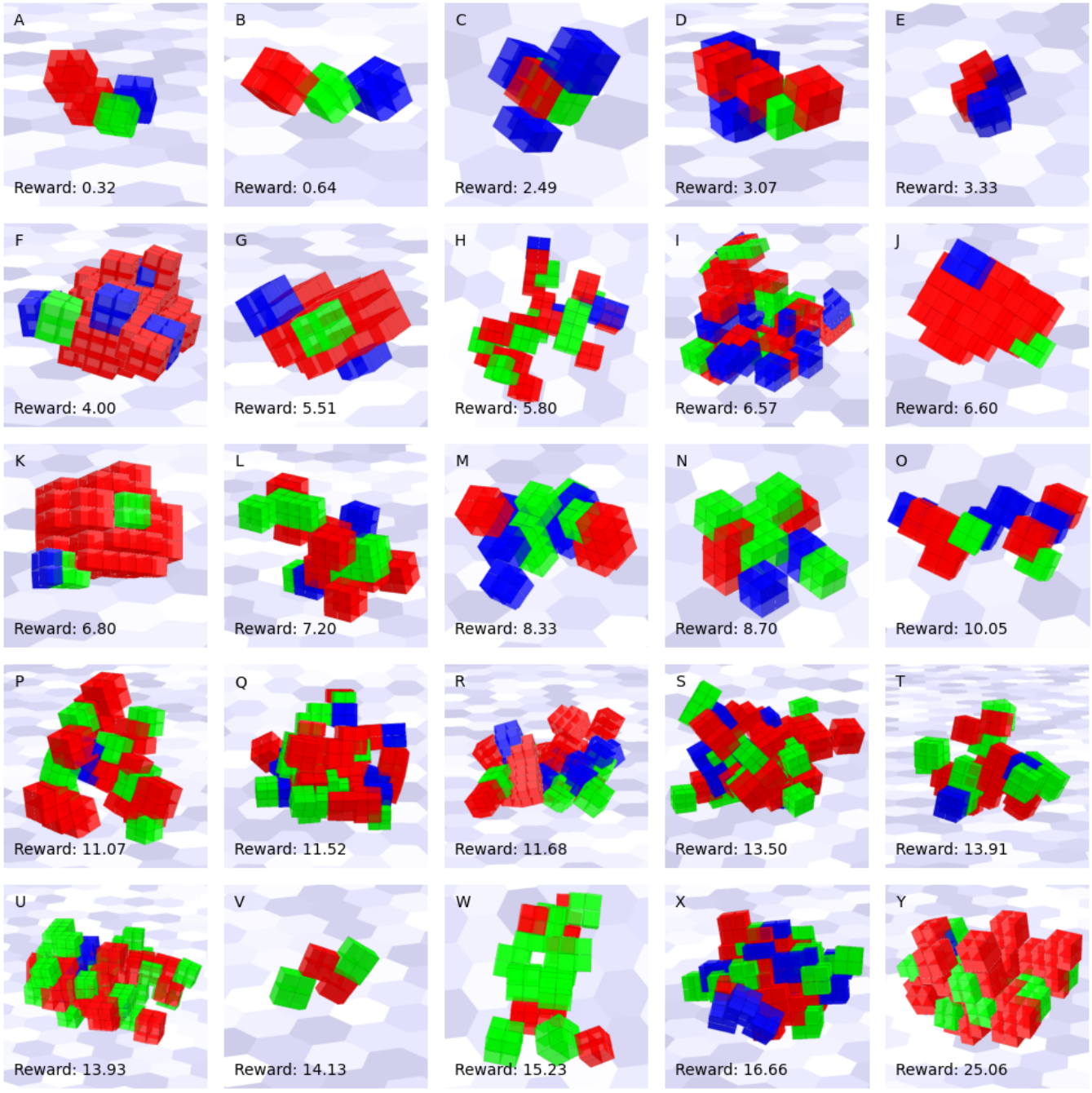

Fig. 3. **Designs for locomotion** sampled across 5 independent trials at different epochs across training. Reward is net displacement (in voxel lengths) measured from evaluation start to end.

$\pi(a_t|s_t) \sim \mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \mathbf{1})$. When $a_t \sim \mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \mathbf{1})$ is mapped to voxel space through the clipping function $f(a)$, approximately 95% of placed voxel bundle centers fall withing the grid $\mathbf{G}$, while the rest fall onto the boundary. Hyperparameters are listed in Table I; unmentioned parameters used the default configuration provided in Version 2.0.1 of Ray [20].

III. Results

We begin by quickly testing our design pipeline to ensure everything is working properly. To do so, we trained the policy for 500 epochs against a simple structural goal: maximize body volume using a single material (i.e. $k = 2$; Fig. 2). We repeated this experiment four times under different random seeds, yielding five independent trials (Fig. 2). The trained policy consistently learned to produce large bodies as evidenced by the significantly smaller bodies produced by the untrained policy ($p < 0.01$). The untrained policy tends to scatter voxel bundles throughout the workspace with just 11.62% of the voxels within the workspace, $\mathbf{G}$, contributing to the body $\mathcal{B}$,

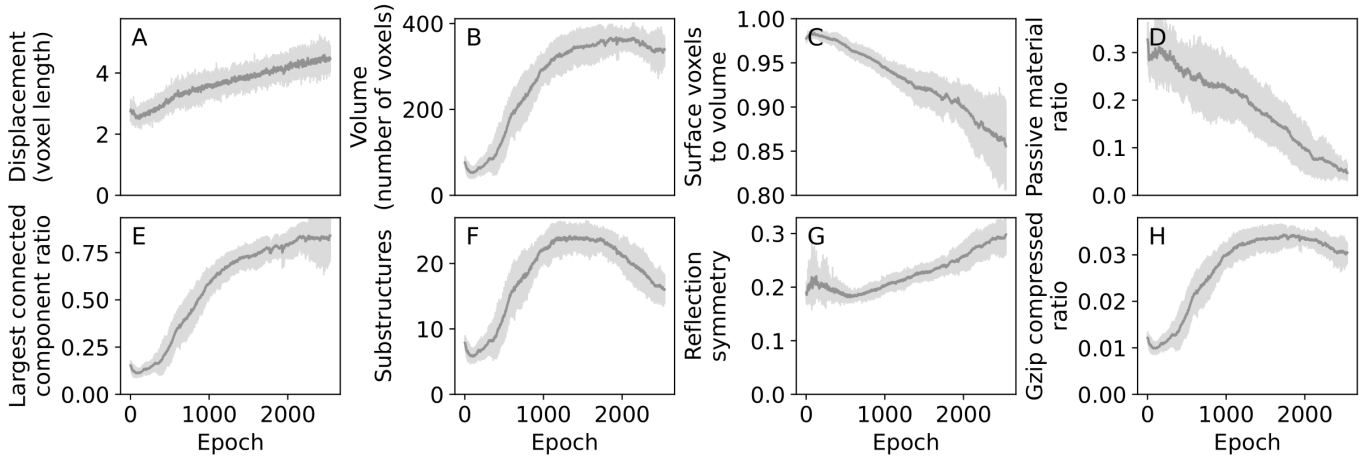

Fig. 4. **Learning to design freeform robots.** Mean (dark gray curve) and 99% Normal confidence intervals (light gray bands) of reward (net displacement in voxel lengths; **A**), body volume (number of voxels; **B**), surface voxels to volume ratio (**C**), passive material ratio (**D**), largest connected component ratio (**E**), number of substructures (separate material regions; **F**), reflection symmetry (**G**), and compressability (using gzip; **H**) during policy optimization across 5 independent learning trials. The policy learned to produce larger, more symmetrical bodies with less passive tissue with higher complexity as measured by the number of substructures and compression score.

on average. Over the course of training, the policy gradually learns to more efficiently utilize its actions such that, by the end of training, the vast majority (82.62%) of voxels in $\mathbf{G}$ were also in $\mathcal{B}$.

Seeing that the policy was able to learn to design large coherent bodies we turned to the optimization of self-moving bodies: robots. The reward was taken to be the maximum euclidean distance in the horizontal plane between the start and end position of each voxel in the robot, measured in voxel lengths, across a 5 sec evaluation period (42K timesteps with stepsize 0.000118 sec), which followed a 5 sec burn-in (42K timesteps with stepsize 0.000118 sec). For this design task, we once again conducted five independent trials, each with their own unique random seed. Each trial optimized a batch of 128 designs on a computer with either two A100 or three RTX A6000 GPUs for ∼28 hours.

The policy, which was trained from scratch in each trial, learned to design better robots, and the trained critic $V(s)$, which was also trained from scratch, learned the concept of locomotion, as evidenced by the significantly less motile bodies produced by the untrained policy (Fig. 4A; $p < 0.01$) and the significantly lower accuracy of the untrained critic (Fig. 5; $p < 0.01$), respectively.

A random sample of robots designed by the policy over the course of training can be seen in Fig. 3. The bodies with high rewards generally possess three design principles: (i) a relatively low passive material ratio (Fig. 3V,W,Y); (ii) multiple structures consisting of active materials in contact with the surface plane (Fig. 3U-Y); and (iii) a relatively high volume, which indirectly increases the number of active surface contacts (Fig. 3S,U,X,Y). The model's discovery of these design principles is reflected in the body metrics shown in Fig. 4. The training generally increases the robot's body size (Fig. 4B) while decreasing its surface area to volume ratio (Fig. 4C), and passive material ratio (Fig. 4D). This is accomplished by cohering the initially scattered design actions into a unified body (Fig. 4E) with more distinct substructures (Fig. 4F), symmetry (Fig. 4G), and overall complexity as measured by the Gzip compression ratio (Fig. 4H).

Reflection symmetry was calculated as the average ratio of voxels which remain unchanged when the body is mirrored across the three center planes of the robot's bounding box.

To determine the robustness of the optimized designs, we simulated the 128 robots from the final batch in each independent trial, this time simulating the robot after each

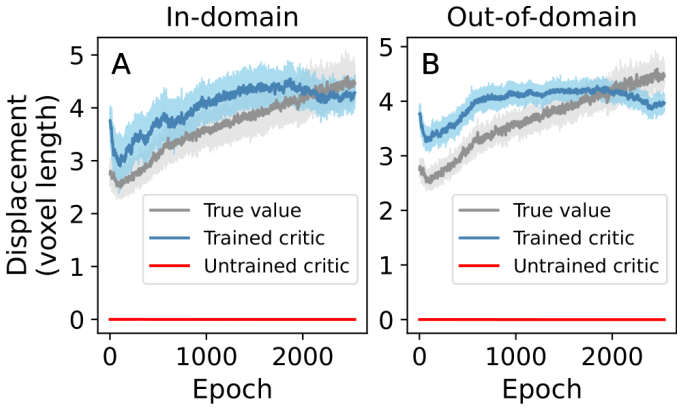

Fig. 5. **Critic estimation of behavioral reward.** Predicted behavioral reward of the untrained (red) and trained (blue) critic against ground truth (gray) for designs generated at each epoch during training. Colored bands denote 99% Normal confidence intervals across the 5 independent trials. The trained critic has learned the concept of locomotion as demonstrated by a significant improvement in prediction ability over the untrained critics. In-domain predictions are computed from bodies that each critic has seen during training (**A**), and out-of-domain predictions are computed from bodies taken from the sibling trials that each critic was not trained under (**B**). The trained critics generalize well to out-of-domain bodies, providing evidence that their understanding of the concept of locomotion extends beyond the specific bodies they saw during training.

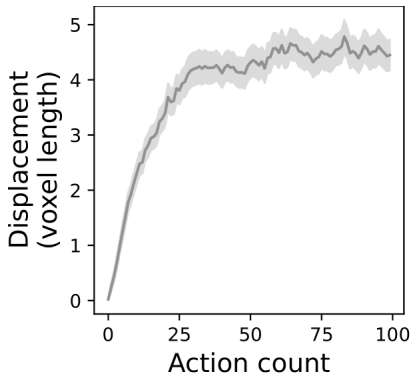

Fig. 6. **Robustness.** Robots sampled from the final epoch of training were simulated at each intermediate design step $t = 1, 2 \ldots 99$. Behavioral reward of robots appears to converge within the first 30 actions.

TABLE I
HYPERPARAMETERS USED FOR VOLUME AND LOCOMOTION TASKS.

Hyperparameter	Value
CNN Size	(3x3x3, 4 kernels) (5x5x5, 1 kernel)
Actor MLP Size	Vol: (128, 128) / Loco: (256, 256)
Critic MLP Size	Vol: (128, 128) / Loco: (256, 256)
PPO Learning Rate	1e-4
PPO Batch Size	Vol: 25600 / Loco: 12800
PPO SGD Minibatch Size	128
PPO SGD iter per Minibatch	Vol: 50 / Loco: 10
PPO Training Epochs	Vol: 500 / Loco: 2500
PPO Discount Factor	0.99
PPO Critic Parameter Clipping	Disabled (by setting to 1e5)
PPO GAE	Disabled

design action instead of just after the last action $t = 100$ (Fig. 6). The mean reward of these partially constructed robots quickly converges within the first 30 design actions, which suggests that the additional 70 actions taken by the policy may be unnecessary.

IV. Discussion

In this paper, we showed that policy gradients can, under certain conditions, admit *de novo* optimization of nonparametric body plans. However there were several limitations and assumptions that may be relaxed by future studies. For example, the kind, size and shape of building blocks made available to the policy, as well as the coordinate system in which the policy acts, were presupposed and held fixed throughout optimization. Also, reward was based solely on the behavior of the whole machine, without supplying reward signals *during* body building. Both of these limitations seem quite unnatural as animal architectures are organized hierarchically and take shape under rich and continual environmental feedback. Indeed, while the employed design algorithm, reward function, and encoding (Fig. 1) performed reasonably well for the structural goal we tested (volume; Fig. 2), optimization was relatively slow for the tested behavioral goal (forward locomotion) as indicated by the average reward curve (Fig. 4A) which had yet to converge after 28 hours. But the most important limitation of the results is that they were provided for simulated robots only.

Future work will involve transferring simulated voxelized machines to physical ones [12, 18, 31] using sim2real methods that avoid difficult-to-simulate structures [16] and dynamics [15], or those that modify the simulator itself [10], based on discrepancies between predicted (sim) and actual (real) behavior. Simulator calibration, as from neural-augmentation of the underlying physics model [10], could take place during morphological pretraining or it may be applied as a transferability filter just before finetuning. The simulator would also be improved by scaling the number of voxels [1] and implementing adaptive resolution grids to efficiently distribute these resources in space and time. For example, it may be beneficial to resolve surface contact geometries at a finer granularity than internal materials, or to gradually increase global resolution over the course of training. But this will ultimately depend on the robot's intended niche.

Finally, it is important to note that while the process of generating a body was mediated by the environment, the behavior of the resulting robots was not. Locomotion was instead coordinated by central pattern generators [14] and "steered" by the robot's overall shape and motor layout. If sensor voxels are added to the design space, the policy could learn how to pattern receptors throughout the robot's body in order to capture relevant stimuli for transduction. For instance, macrostructures built out of mechanoreceptive voxels [34] could, depending on their arrangement, "listen to" certain load signatures, and ignore others. The emergence of more complex sensory organs and sense-guided behaviors could be hastened by predator and prey scenarios [25], if they trigger a co-evolutionary arms race of morphological adaptations.

Code

The source code necessary to reproduce the results presented in this paper can be found with MIT License on GitHub: **github.com/iffiX/RL4design**

Acknowledgments

We thank V.S. Subrahmanian for lending computational resources that enabled this research, and Chris Fusting for helpful discussions. This work was supported in part by the AI2050 program at Schmidt Sciences (Grant G-22-64506) and a seed grant from the Center for Engineering Sustainability and Resilience at Northwestern University.

References

[1] Niels Aage, Erik Andreassen, Boyan S Lazarov, and Ole Sigmund. Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674):84–86, 2017.

[2] Josh Bongard. The utility of evolving simulated robot morphology increases with task complexity for object manipulation. *Artificial Life*, 16(3):201–223, 2010.

[3] Josh Bongard and Rolf Pfeifer. Evolving complete agents using artificial ontogeny. In *Morpho-functional machines: the new species: designing embodied intelligence*, pages 237–258, 2003.

[4] Nick Cheney, Josh Bongard, Vytas SunSpiral, and Hod Lipson. Scalable co-optimization of morphology and control in embodied machines. *Journal of The Royal Society Interface*, 15(143):20170937, 2018.

[5] François Cochevelou, David Bonner, and Martin-Pierre Schmidt. Differentiable soft-robot generation. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, page 129–137. Association for Computing Machinery, 2023.

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[7] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature Communications*, 12(1):1–12, 2021.

[8] David Ha. Reinforcement learning for improving agent design. *Artificial Life*, 25(4):352–365, 2019.

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

[10] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S Sukhatme. Neuralsim: Augmenting differentiable simulators with neural networks. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 9474–9481, 2021.

[11] Donald J Hejna III, Pieter Abbeel, and Lerrel Pinto. Task-agnostic morphology evolution. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[12] Jonathan Hiller and Hod Lipson. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466, 2012.

[13] Jonathan Hiller and Hod Lipson. Dynamic simulation of soft multimaterial 3D-printed objects. *Soft Robotics*, 1 (1):88–101, 2014.

[14] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, 21(4):642–653, 2008.

[15] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *Proceedings of the European Conference on Artificial Life (ECAL)*, pages 704–720, 1995.

[16] Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 119–126, 2010.

[17] Sam Kriegman, Douglas Blackiston, Michael Levin, and Josh Bongard. A scalable pipeline for designing reconfigurable organisms. *Proceedings of the National Academy of Sciences*, 117(4):1853–1859, 2020.

[18] Sam Kriegman, Amir Mohammadi Nasab, Dylan Shah, Hannah Steele, Gabrielle Branin, Michael Levin, Josh Bongard, and Rebecca Kramer-Bottiglio. Scalable sim-to-real transfer of soft robot designs. In *Proceedings of the International Conference on Soft Robotics (RoboSoft)*, pages 359–366, 2020.

[19] Sam Kriegman, Douglas Blackiston, Michael Levin, and Josh Bongard. Kinematic self-replication in reconfigurable organisms. *Proceedings of the National Academy of Sciences*, 118(49):e2112672118, 2021.

[20] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 3053–3062, 2018.

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

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

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

[24] Ugurcan Mugan and Malcolm A MacIver. Spatial planning with long visual range benefits escape from visual predators in complex naturalistic environments. *Nature Communications*, 11(1):1–14, 2020.

[25] Stefano Nolfi and Dario Floreano. Coevolving predator and prey robots: Do "arms races" arise in artificial evolution? *Artificial life*, 4(4):311–335, 1998.

[26] Deepak Pathak, Christopher Lu, Trevor Darrell, Phillip Isola, and Alexei A Efros. Learning to control self-assembling morphologies: a study of generalization via modularity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[27] Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to construct and control agents using deep reinforcement learning. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 9798–9805, 2019.

[28] Charles Schaff, Audrey Sedal, and Matthew R Walter. Soft robots learn to crawl: Jointly optimizing design and control with sim-to-real transfer. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[30] Karl Sims. Evolving 3D morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.

[31] Mark A Skylar-Scott, Jochen Mueller, Claas W Visser, and Jennifer A Lewis. Voxelated soft matter via multimaterial multinozzle 3D printing. *Nature*, 575(7782): 330–335, 2019.

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

[33] Justin K Terry, Nathaniel Grammel, Ananth Hari, Luis Santos, and Benjamin Black. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625*, 2020.

[34] Ryan L Truby, Lillian Chin, Annan Zhang, and Daniela Rus. Fluidic innervation sensorizes structures from a single build material. *Science Advances*, 8(31):eabq4385, 2022.

[35] Yuxing Wang, Shuang Wu, Haobo Fu, Qiang Fu, Tiantian Zhang, Yongzhe Chang, and Xueqian Wang. Curriculum-based co-design of morphology and control of voxel-based soft robots. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

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

[37] Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris Kitani. Transform2act: Learning a transform-and-control policy for efficient agent design. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[38] Changyoung Yuhn, Yuki Sato, Hiroki Kobayashi, Atsushi Kawamoto, and Tsuyoshi Nomura. 4d topology optimization: Integrated optimization of the structure and self-actuation of soft bodies for dynamic motions. *Computer Methods in Applied Mechanics and Engineering*, 414:116187, 2023.

[39] Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Robogrammar: graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.