Morphology-Adaptive Muscle-Driven Locomotion via Attention Mechanisms

Junior Rojas
jrojasdavalos@gmail.com

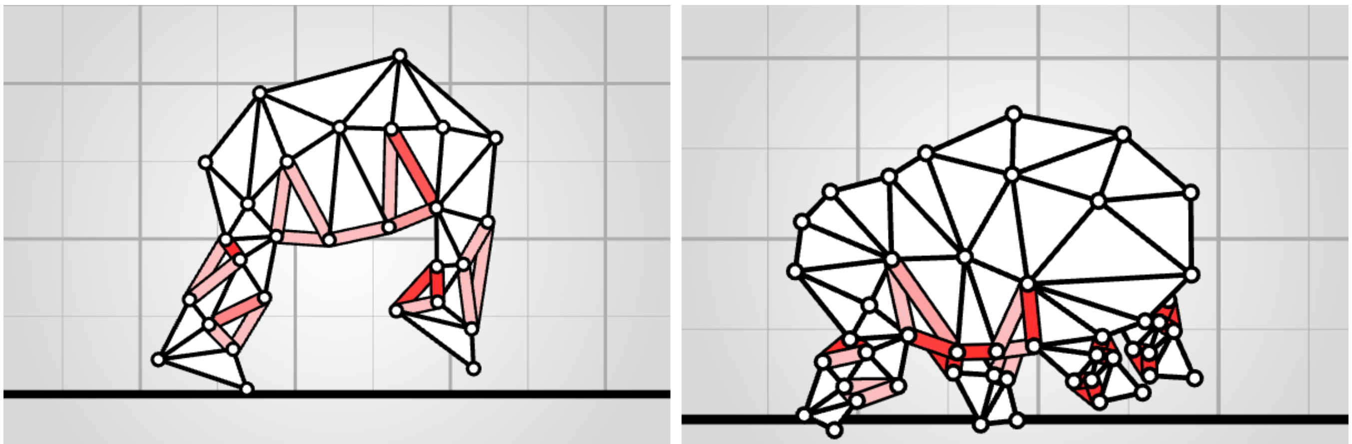

Figure 1: The same attention-based policy can control both a bipedal (left) and a quadrupedal (right) body plan. Video available at https://www.youtube.com/watch?v=gmgyFIJz9ZY.

Abstract

This paper introduces an attention-based approach to muscle-driven locomotion that allows a single controller to adapt to different body plans. We consider physically simulated agents with bodies modeled as deformable meshes and controllable fibers that act as muscles. Vertices serve as addressable memory, defining keys and values, while muscles generate multi-head queries to extract relevant information from vertex states and produce actuation signals. A key characteristic of our approach is that all muscle fibers share the same attention-based control policy. This enables locomotion controllers that accommodate any number of inputs and outputs while also remaining permutation invariant. We demonstrate the effectiveness of our method by successfully controlling both bipedal and quadrupedal body plans using the same control policy.

CCS Concepts

• **Computing methodologies** → **Neural networks**.

Keywords

Neural Networks, Attention Mechanisms, Virtual Creatures

GECCO '25 Companion, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3734277

ACM Reference Format:
Junior Rojas. 2025. Morphology-Adaptive Muscle-Driven Locomotion via Attention Mechanisms. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3712255.3734277

1 Introduction

Control policies for locomotion are typically designed for a specific morphology with a rigid input-output structure. Modifications such as reordering inputs and outputs or adding and removing elements can render a policy unusable. In contrast, biological processes like metamorphosis and regeneration demonstrate how functionality and memory can persist through major structural transformations [1]. During metamorphosis, the brain and body are extensively rebuilt, yet memories persist and are remapped to a completely new sensorimotor configuration [2]. This raises fundamental questions about how past experiences are compressed into a form that can survive radical transformations and be reinstantiated in a new body.

As a step towards addressing this challenge in the context of muscle-driven locomotion, we propose an approach that adapts to different body plans by using a shared policy across all muscle actuators. We consider agents modeled as deformable meshes with controllable fibers, an approach that has proven effective for muscle-based locomotion [10, 19]. In contrast to prior control approaches that rely on neural networks with fixed input-output structures, our method supports a variable number of inputs and outputs while remaining permutation invariant. This property, sometimes referred to as structural flexibility, has been highlighted as a key enabler of adaptability across environments in general reinforcement learning settings [15, 22].

In the context of locomotion specifically, there is growing interest in morphology-adaptive control and the joint optimization of morphology and control [4, 6, 7, 9, 12, 14, 16, 21, 24–26], highlighting the importance of developing controllers that can generalize across different body plans. One of the main goals of our work is to explore models that can accommodate significantly different morphologies while keeping architectural complexity low. Attention mechanisms are particularly well suited for this task, as they naturally handle a variable number of inputs and enable permutation invariance.

While attention mechanisms have gained widespread adoption due to the success of transformers in language models [23], the language model perspective, which assumes a sequential processing order, is not the most relevant in our setting. Unlike attention-based approaches in language models, where the context expands as new tokens are generated, our model maintains a fixed temporal context. Decisions at each time step depend only on the current vertex positions and velocities. The primary challenge is not handling expanding temporal dependencies, but rather integrating information across a variable number of spatially distributed nodes.

A more relevant analogy to early work in attention models can be found in Neural Turing Machines (NTMs) [3]. NTMs use an attention mechanism to selectively read from an addressable memory, and then write back to it after processing it. While the specific architectural details differ, this provides a valuable mental model for our approach: vertices act as addressable memory, while muscles act as write heads, influencing the updated vertex states.

2 Morphology-Specific Control

This work uses the open-source project Algovivo [17], which implements an energy-based approach for muscle-driven locomotion, where system dynamics are governed by the minimization of six differentiable functions. Muscle fibers embedded in a deformable mesh act as agents seeking states of minimal elastic potential energy, and locomotion is achieved by reshaping their energy landscape [18].

Muscle fibers are modeled as springs with parameters from Hookean elasticity, including rest length l_0 and stiffness k, with modifications to accommodate actuation. A scaling factor a, where $a \leq 1$, modulates the effective rest length, enabling the muscle to contract relative to its original rest length. The muscle's current length l depends on the vertex positions p and the energy of the muscle is defined as a function of p and a:

$$E(p, a) = \frac{k}{2}\left(\frac{l(p)}{al_0} - 1\right)^2 \tag{1}$$

Changes in a over time generate different locomotion patterns, as muscles continuously adjust toward the shifting local minima defined by a. However, even when a is fixed, the muscle length does not necessarily settle at its effective rest length al_0, since the positions of the connected vertices affect other energy terms of the system. The deformable mesh is composed of neo-Hookean elastic triangles, which resist large deformations and contribute to shape recovery, providing stability that springs alone cannot achieve [20]. This competition between energy terms determines the final configuration after energy minimization.

A locomotion control policy changes a over time based on sensory input. Prior work commonly uses multilayer perceptrons (MLPs) to map vertex positions and velocities (which can be interpreted as proprioceptive and kinesthetic signals) to muscle actuation signals [19]. This method has been effective for locomotion and commonly includes a frame projection step to enforce translation and rotation invariance, as illustrated in Figure 2.

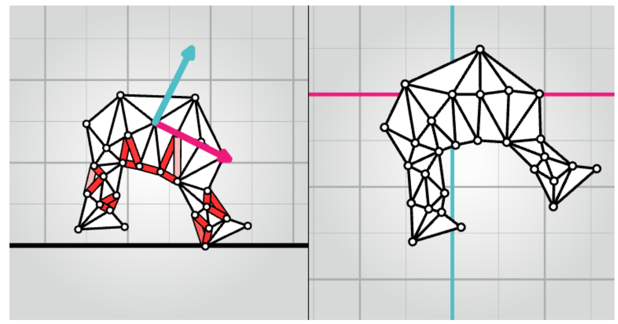

Figure 2: Mesh in world space (left) and after projection onto the agent's local coordinate frame (right). In the morphology-specific approach, an MLP maps vertex positions and velocities to muscle actuation signals. Vertex positions and velocities are projected onto a local frame before being sent to the MLP to enforce translation and rotation invariance. The local frame is defined by two reference vertices: a center vertex and a forward vertex, which determine the forward direction. The up direction is computed as the orthogonal direction, counterclockwise from the forward direction. Muscle control signals reshape the energy landscape by adjusting the effective rest length of each muscle, generating locomotion patterns as muscles continuously adjust toward the shifting local minima defined by the energy function.

3 Morphology-Adaptive Attention Mechanism

In an MLP, inputs and outputs are fixed in number and order, meaning each connection of the network is set to work with a very specific input-output structure, making the policy unusable if vertices or muscles are added, removed, or reordered. To enable adaptability to varying numbers of inputs, we use an attention mechanism where inputs are accessed through keys rather than a fixed ordering. To enable adaptability to varying numbers of outputs, we use a shared policy across all contractile fibers, allowing the same controller to actuate any number of muscles.

Each vertex has a position and velocity that evolve over time during the simulation, which serve as the values in the attention mechanism. To ensure translation and rotation invariance, we also normalize these quantities using a local frame before passing them through the attention mechanism.

The key of each vertex is defined by its position in the rest pose, which remains constant over time and is determined by the body plan. While vertices themselves do not inherently have a rest location, muscles have a rest length (used to compute the Hookean elastic energy) and triangles have a rest shape (used to compute the neo-Hookean elastic energy). In practice, these rest

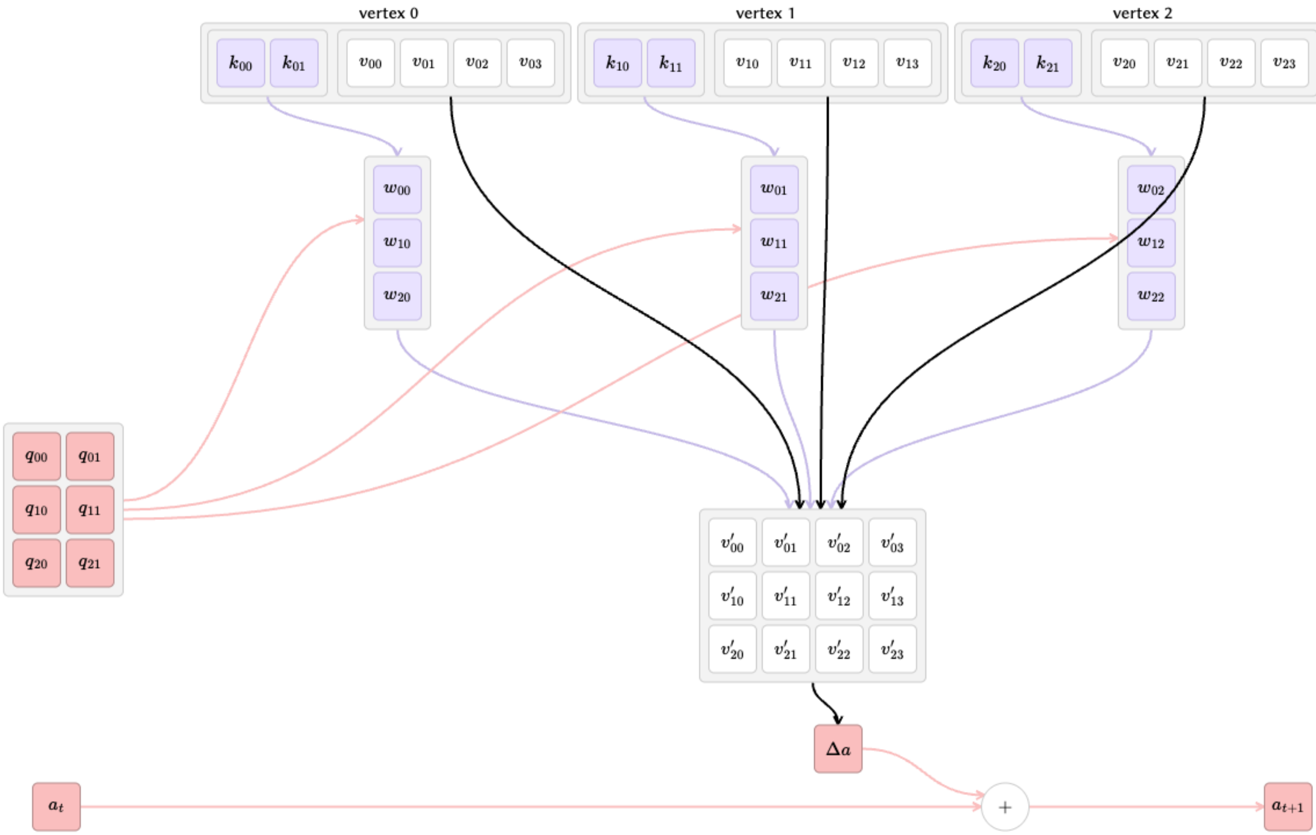

Figure 3: Attention mechanism for muscle control. Each muscle generates a multi-head query q, which is used to compute a weighted sum v' of vertex values. The figure illustrates an example with 3 attention heads, 3 vertices, a key size of 2, and a value size of 4 per vertex. The resulting weighted values are processed by an MLP, which updates the actuation signal at each time step as $a_{t+1} = a_t + \Delta a$.

quantities are derived from a rest pose defined by vertex positions [20]. Translating or rotating the rest pose does not alter the rest lengths of muscles or the rest shapes of triangles. To ensure the keys remain consistent under these transformations, we also normalize vertex positions in the rest pose using a local frame.

Each muscle is identified by its position in the rest pose, computed as the midpoint of the two vertices it connects (Figure 4). This positional reference serves as the muscle identifier, conceptually similar to a key. However, since this identifier is used to query the vertex-based memory, it is first transformed by an MLP into a multi-head query for the attention mechanism.

To better understand how muscles extract relevant information from the mesh, we break down the computations performed per muscle shown in Figure 3. Consider a multi-head query $q \in \mathbb{R}^{H \times d_k}$, generated by a muscle, where H is the number of heads and d_k is the key size. Consider vertex keys $k \in \mathbb{R}^{V \times d_k}$ and vertex values $v \in \mathbb{R}^{V \times d_v}$, where V is the number of vertices and d_v is the value size. In our particular case, $d_v = 4$ (position and velocity in 2D) and $d_k = 2$ (location in the 2D rest pose). For each head, the attention mechanism computes weights $w \in \mathbb{R}^V$ based on the similarity of the keys and the query. First, we compute unnormalized weights

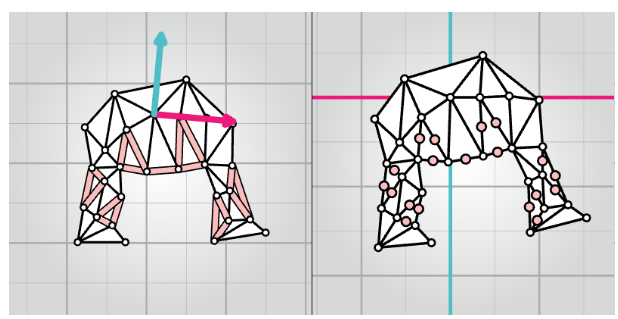

Figure 4: Rest pose in world space (left) and after projection onto the agent's local coordinate frame, with muscle nodes shown (right). Each muscle is identified by its position in the rest pose (pink nodes), computed as the midpoint of the two vertices it connects. This positional reference serves as the muscle identifier, which is transformed by an MLP into a multi-head query. The query is then used to extract relevant information from the vertex states via attention.

$u \in \mathbb{R}^{H\times V}$, where the i-th unnormalized weight of the k-th head (u_{ki}) is the dot product of the k-th head of the query (q_k) and the key of the i-th vertex (k_i):

$$u_{ki} = \sum_{j}^{d_k} q_{kj} k_{ij} \tag{2}$$

The unnormalized weights u are normalized using a softmax function to obtain the final weights w, and then used to compute v', the weighted sum of the values:

$$\underbrace{w_k}_{\mathbb{R}^V} = \text{softmax}(\underbrace{u_k}_{\mathbb{R}^V}), \quad \underbrace{v'_k}_{\mathbb{R}^{d_v}} = \sum_{i}^{V} w_{ki} \underbrace{v_i}_{\mathbb{R}^{d_v}} \tag{3}$$

At each time step, each muscle computes its own $v' \in \mathbb{R}^{H\times d_v}$ based on the current vertex positions and velocities. Then, v' is passed through an MLP to produce the final muscle actuation signal. As in prior work [19], instead of directly outputting a, the MLP outputs Δa, which updates the muscle control parameter for the next simulation step as $a_{t+1} = a_t + \Delta a$ (Figure 3). This update continuously reshapes the system's energy landscape, driving locomotion.

4 Bipedal and Quadrupedal Locomotion

To test the effectiveness of our method, we trained a single policy for both bipedal and quadrupedal locomotion (Figure 1). Our focus is not on learning locomotion from scratch but on unifying different body plans and their locomotion behaviors into a single policy using attention-based control. We used reference trajectories generated by previously trained MLP policies, available from prior work [17, 18], each trained for a specific morphology. The bipedal body plan consists of 28 vertices, 19 muscle fibers and 35 triangles, while the quadrupedal body plan consists of 62 vertices, 38 muscle fibers and 95 triangles.

Training was performed using the Adam optimizer [8] with a standard supervised learning loop in PyTorch [13]. Figure 5 shows one training run using 150 steps of bipedal locomotion and 150 steps of quadrupedal locomotion as training data. While this procedure can produce behavior similar to that of the reference policies, we observed occasional deviations from the original behavior when running the trained policy. These deviations are likely due to the reliance on a single ideal trajectory during training, which makes the policy brittle when encountering states that differ significantly from those seen during training. We found that augmenting the training set with new reference trajectories initialized from the problematic states improved the policy in subsequent training runs.

5 Discussion and Future Work

Our approach is both modular and decentralized, as muscle fibers act as reusable, independent processing units that share the same control policy. Unlike MLP-based policies, which require a predefined number of outputs and rely on a central hidden layer to generate all muscle actuation signals, our method assigns each muscle its own attention-based processing mechanism. This design enables the controller to adapt to different body plans, accommodating any number of inputs and outputs.

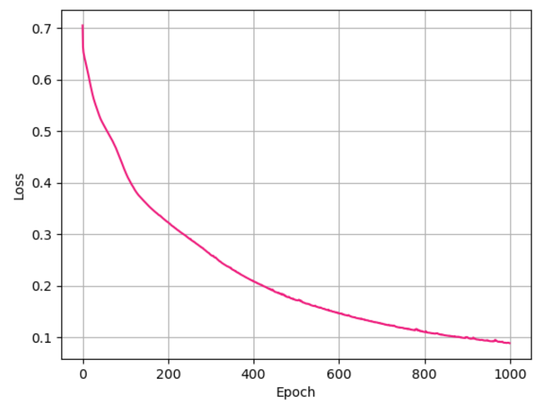

Figure 5: Training progress using 150 steps of bipedal locomotion and 150 steps of quadrupedal locomotion, each taken from a single ideal reference trajectory. While this simple setup illustrates the attention-based policy's ability to imitate behavior across different body plans, additional trajectories initialized from different states are often needed to improve robustness.

Modular and decentralized control is often associated with local interactions, where each module processes only nearby elements within a limited sensory range [7, 12, 14]. However, prior work has also demonstrated modular control of soft robots without explicit inter-module communication [16]. Our approach lies at the opposite end of this spectrum, where each module has a broad sensory range, considering all vertices in the mesh as potential inputs and selectively extracting relevant vertex states through attention.

The meshes used in our experiments define two reference vertices to establish a local coordinate system, ensuring translation and rotation invariance. While this choice was made for consistency with prior work [19], it is not a fundamental requirement, and alternative sensory input representations could be explored. The key aspect of our approach is that muscles extract relevant information through attention over a key-value format of the sensory input.

Future work could explore localized attention, where elements process only a subset of mesh elements and propagate information iteratively. Since our approach accommodates variable input sizes, it naturally supports arbitrary meshes, where elements can vary in both size and connectivity, each having a non-uniform number of neighbors.

Many modular and decentralized control approaches rely on regular grids, such as neural cellular automata and voxel-based soft robots [5, 11, 12]. These approaches have produced interesting results, and the simplicity of regular grids, along with the availability of convolution operators in modern software libraries, makes them easy to implement and experiment with.

To further advance the field, we see great potential in exploring more irregular representations, such as the mesh-based structures considered in this work. Meshes allow for coarser models that capture essential morphological features with fewer elements while also being able to represent regular grids as a special case.

Many distributed and decentralized control approaches draw inspiration from biological processes, such as morphogenesis and other collective cell behaviors. While regular grids sometimes offer computational convenience, they are a significant abstraction away from the irregular interactions seen in cell collectives. Non-uniform spatial arrangements and attention mechanisms are particularly well suited for modeling such irregularities.

This work represents a step in that direction. We trained a minimal architecture to demonstrate that attention-based control can achieve similar performance as MLP policies for two distinct body plans. While our current setup is limited to 2D environments and supervised training based on reference trajectories, the architecture can be extended to 3D environments using tetrahedral meshes and adapted to training loops that optimize directly for locomotion by continuously interacting with the environment. This could help address the limitations of imitation-based training, which may cause the policy to fail when the agent encounters previously unseen states, unless additional data is introduced, as discussed in Section 4. We see this work as a starting point for future developments, including broader morphological generalization, alternative training methods, and architectural enhancements.

References

[1] Douglas Blackiston, Tal Shomrat, and Michael Levin. 2015. The stability of memories during brain remodeling: A perspective. *Communicative & Integrative Biology* 8, 5 (2015), e1073424. doi:10.1080/19420889.2015.1073424 arXiv:https://doi.org/10.1080/19420889.2015.1073424 PMID: 27066165.

[2] Douglas J. Blackiston, Elena Silva Casey, and Martha R. Weiss. 2008. Retention of Memory through Metamorphosis: Can a Moth Remember What It Learned As a Caterpillar? *PLOS ONE* 3, 3 (03 2008), 1–7. doi:10.1371/journal.pone.0001736

[3] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing Machines. arXiv:1410.5401 [cs.NE] https://arxiv.org/abs/1410.5401

[4] Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. 2022. MetaMorph: Learning Universal Controllers with Transformers. In *International Conference on Learning Representations*.

[5] Kazuya Horibe, Kathryn Walker, and Sebastian Risi. 2021. Regenerating Soft Robots Through Neural Cellular Automata. In *Genetic Programming: 24th European Conference, EuroGP 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings*. Springer-Verlag, Berlin, Heidelberg, 36–50. doi:10.1007/978-3-030-72812-0_3

[6] Suning Huang, Boyuan Chen, Huazhe Xu, and Vincent Sitzmann. 2024. DittoGym: Learning to Control Soft Shape-Shifting Robots. In *The Twelfth International Conference on Learning Representations*.

[7] Wenlong Huang, Igor Mordatch, and Deepak Pathak. 2020. One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. In *ICML*.

[8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[9] Muhan Li, Lingji Kong, and Sam Kriegman. 2025. Generating Freeform Endoskeletal Robots. In *The Thirteenth International Conference on Learning Representations*.

[10] Sehee Min, Jungdam Won, Seunghwan Lee, Jungnam Park, and Jehee Lee. 2019. SoftCon: simulation and control of soft-bodied animals with biomimetic actuators. *ACM Trans. Graph.* 38, 6, Article 208 (Nov. 2019), 12 pages. doi:10.1145/3355089.3356497

[11] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. 2020. Growing Neural Cellular Automata. *Distill* (2020). doi:10.23915/distill.00023 https://distill.pub/2020/growing-ca.

[12] Giorgia Nadizar, Eric Medvet, Kathryn Walker, and Sebastian Risi. 2023. A fully-distributed shape-aware neural controller for modular robots. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 184–192.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

[14] Deepak Pathak, Chris Lu, Trevor Darrell, Phillip Isola, and Alexei A. Efros. 2019. Learning to Control Self-Assembling Morphologies: A Study of Generalization via Modularity. In *NeurIPS*.

[15] Joachim Pedersen, Erwan Plantec, Eleni Nisioti, Milton Montero, and Sebastian Risi. 2024. Structurally Flexible Neural Networks: Evolving the Building Blocks for General Agents. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Melbourne, VIC, Australia) *(GECCO '24)*. Association for Computing Machinery, New York, NY, USA, 1119–1127. doi:10.1145/3638529.3654113

[16] Federico Pigozzi, Yujin Tang, Eric Medvet, and David Ha. 2022. Evolving modular soft robots without explicit inter-module communication using local self-attention. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 148–157. doi:10.1145/3512290.3528762

[17] Junior Rojas. 2023. *Algovivo: An energy-based formulation for soft-bodied virtual creatures*. https://github.com/juniorrojas/algovivo

[18] Junior Rojas. 2024. Energy-Based Models for Virtual Creatures. ALIFE 2024: Proceedings of the 2024 Artificial Life Conference (07 2024), 30. doi:10.1162/isal_a_00748 arXiv:https://direct.mit.edu/isal/proceedings-pdf/isal2024/36/30/2461221/isal_a_00748.pdf

[19] Junior Rojas, Stelian Coros, and Ladislav Kavan. 2019. Deep reinforcement learning for 2D soft body locomotion. In *NeurIPS Workshop on Machine Learning for Creativity and Design 3.0*.

[20] Eftychios Sifakis and Jernej Barbic. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California) *(SIGGRAPH '12)*. Association for Computing Machinery, New York, NY, USA, Article 20, 50 pages. doi:10.1145/2343483.2343501

[21] Luke Strgar, David Matthews, Tyler Hummer, and Sam Kriegman. 2024. Evolution and learning in differentiable robots. arXiv:2405.14712 [cs.RO] https://arxiv.org/abs/2405.14712

[22] Yujin Tang and David Ha. 2021. The Sensory Neuron as a Transformer: Permutation-Invariant Neural Networks for Reinforcement Learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*. https://attentionneuron.github.io.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[24] Tsun-Hsuan Wang, Juntian Zheng, Pingchuan Ma, Yilun Du, Byungchul Kim, Andrew Everett Spielberg, Joshua B. Tenenbaum, Chuang Gan, and Daniela Rus. 2023. DiffuseBot: Breeding Soft Robots With Physics-Augmented Generative Diffusion Models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[25] Julian Whitman, Matthew Travers, and Howie Choset. 2023. Learning Modular Robot Control Policies. *IEEE Transactions on Robotics* 39, 5 (2023), 4095–4113. doi:10.1109/TRO.2023.3284362

[26] Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris Kitani. 2022. Transform2Act: Learning a Transform-and-Control Policy for Efficient Agent Design. In *International Conference on Learning Representations*.