Multi-embodiment Legged Robot Control as a Sequence Modeling Problem

Chen Yu[1,2], Weinan Zhang[3], Hang Lai[2,3], Zheng Tian[4], Laurent Kneip[1], and Jun Wang[2,5]

Abstract— Robots are traditionally bounded by a fixed embodiment during their operational lifetime, which limits their ability to adapt to their surroundings. Co-optimizing control and morphology of a robot, however, is often inefficient due to the complex interplay between the controller and morphology. In this paper, we propose a learning-based control method that can inherently take morphology into consideration such that once the control policy is trained in the simulator, it can be easily deployed to robots with different embodiments in the real world. In particular, we present the Embodiment-aware Transformer (EAT), an architecture that casts this control problem as conditional sequence modeling. EAT outputs the optimal actions by leveraging a causally masked Transformer. By conditioning an autoregressive model on the desired robot embodiment, past states, and actions, our EAT model can generate future actions that best fit the current robot embodiment. Experimental results show that EAT can outperform all other alternatives in embodiment-varying tasks, and succeed in an example of real-world evolution tasks: stepping down a stair through updating the morphology alone. We hope that EAT will inspire a new push toward real-world evolution across many domains, where algorithms like EAT can blaze a trail by bridging the field of evolutionary robotics and big data sequence modeling.

I. INTRODUCTION

In nature, animal species can exhibit physiological and structural adaptations to changes in environments across multiple generations. This increases their likelihood of survival and the preservation of their genes [1]. However, in the field of robotics—although more and more robots show their ability to evolve their controller through interaction with the real world to improve their adaptivity to the environment [2], [3], [4], [5]—real robots are traditionally bounded by a fixed embodiment during their operational lifetime.

Some previous works in the field of Evolutionary Robotics optimize morphology together with control of robots [6], [7], [8], [9], [10], [11], [12], [13]. These robots and controllers are relatively simple and hence hard to be deployed in real applications. Other works propose hierarchical approaches with two loops: The outer loop evolves morphology while the inner loop optimizes a controller for each new morphology [14], [15], [16], [17], [18]. However, for relatively complicated robots that involve dynamic locomotion, hierarchical approaches often only work in simulation, as it would usually take millions of control steps for evolution [19].

[1] School of Info. Sci. and Tech., ShanghaiTech University, China.
[2] Digital Brain Lab, Shanghai, China
[3] Dept. of Computer Sci. and Eng., Shanghai Jiao Tong University, China.
[4] School of Creativity and Art, ShanghaiTech University, China.
[5] Centre for Artificial Intelligence, University College London, UK.

To tackle these challenges, in this work, we wish to have a controller that inherently takes morphology into consideration. In this way, once a general control policy is trained (e.g., in the simulator), it can be deployed on robots with different embodiments (e.g., in the real world), as shown in Fig. 1. This is usually challenging because of the complicated interplay between robot morphology and control [20].

Fig. 1. **Demonstration of two robot evolution schemes.** We compare the hierarchical approaches (left) and our method based on a general purpose controller (right).

While traditional model-based control approaches for robots—especially with relatively complicated dynamics, such as legged robots—are usually based on analytic dynamics models [21], [22], [23], it is possible to control such a robot with varying morphology by system identification [24], [25]. However, this requires a considerable amount of tedious hand-engineering and a known robot model.

Learning-based methods, such as reinforcement learning (RL), have proven effective at solving an increasing number of real-world locomotion tasks [26], [27], [28]. Chen et al. [29] formulate the policy as a function of the current state and the hardware property encoding. However, it either requires the full kinematics information of the robot, or implicitly learns the hardware representation, in which case it is challenging for zero-shot transfer to unseen robots. Schaff et al. [30] maintain a distribution over designs and use RL to maximize expected rewards over the design distribution. Since the policy optimization is still embedded in a control-morphology co-optimization pipeline, it would be inefficient to update the morphology in the real world. Regarding the morphology as a graph structure, preliminary works also explore this problem through graph neural networks for morphology generalization [31], [32], [33], but none of which is validated in the real world. Concurrently with our work, Feng et al. [34] propose an RL-based general-purpose locomotion controller, GenLoco, using morphology randomization. They validate their controller on three commercially-available robots in the real world.

Note there are recent works formulating RL as a sequence

modeling problem [35], [36]. Decision Transformer [37], [38], [39] uses state, action, and returns-to-go (sum of future rewards) as tokens in a Transformer model. Trajectory Transformer [40] uses a Transformer model to predict the dynamics of a robot and uses beam search [41] for planning. These Transformer-based approaches have achieved similar or better performances in benchmark tasks compared with classic RL algorithms thanks to the model capacity and the self-attention mechanism.

In this work, we also take advantage of Transformer models to design a controller for robots with changing morphology. In particular, we present Embodiment-aware Transformer (EAT), an architecture that casts this control problem as conditional sequence modeling. EAT uses a Transformer architecture to model distributions over trajectories and robot morphology and outputs the optimal actions by leveraging the causally masked Transformer. By conditioning the autoregressive model on the desired robot embodiment (we focus on the morphology in this work), past states, and actions, our EAT model can generate future actions that best fit the current robot embodiment, as shown in Fig. 2.

We validate EAT on a locomotion control task—learning to walk stably from scratch—on the quadruped robot Mini Cheetah [42] both in the simulator and in the real world. We allow the robot to grow its body and leg size to simulate ontogenetic morphological changes. After deploying the trained EAT model on the real robots, we further use Bayesian Optimization (BO) [43] to optimize the morphology of the robots online, while the fixed EAT policy is conditioned on the current body shape.

Experimental results show that EAT can successfully find the controller that fits the current robot morphology. As a consequence, it outperforms all other alternatives in the simulator and succeeds in our real-world evolution task. To the best of our knowledge, this is the first time that a Transformer is applied to an evolutionary robotics task; and this is the first time that real-world morphology evolution is applied to a dynamic quadruped robot.

II. PRELIMINARIES

The Transformer model is introduced by Vaswani et al. [44] for efficient sequential data modeling, which has been shown to perform strongly on various tasks from Natural Language Processing [45], [46] to Computer Vision [47], [48]. It consists of stacked self-attention layers with residual connections. Each self-attention layer maps an input sequence of symbol representations $(x_1, \ldots, x_n)$ with a context length of n to a sequence of continuous representations $\mathbf{z} = (z_1, \ldots, z_n)$. Each token is mapped linearly to a key k_i, query q_i, and value v_i. The corresponding output of the self-attention layer is given by weighting the values v_j by the normalized dot product between the query q_i and other keys k_j:

$$z_i = \sum_{j=1}^{n} \text{softmax}\left(\frac{\{\langle q_i, k_{j'} \rangle\}_{j'=1}^{n}}{\sqrt{d_k}}\right)_j \cdot v_j, \quad (1)$$

where d_k is the dimension of queries and keys.

This can be used for forming state-return associations via similarity of the query and key vectors in the context of offline RL [37]. Offline RL algorithms learn effective policies from previously collected, fixed datasets without further environment interaction [49], [50], [51].

III. EMBODIMENT-AWARE TRANSFORMER

Fig. 2. **Embodiment-aware Transformer architecture.** We learn a linear layer for embodiment, states, and actions for token embeddings, while a positional episodic timestep encoding is added. Tokens are fed into a GPT model that predicts actions autoregressively with a causal self-attention mask.

In this section, we present Embodiment-aware Transformer, as summarized in Fig. 2 and Algorithm 1.

A. Embodiment-aware Markov Decision Process

We model the control of the robot as a variant of Markov decision process (MDP), referred to as embodiment-aware MDP, described by the tuple $(\mathcal{E}, \mathcal{S}, \mathcal{A}, P_E, \mathcal{R})$. The MDP tuple consists of embodiment $e \in \mathcal{E}$, states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, embodiment-dependent state transition dynamics $P_E(\cdot \mid s, a; e)$, and a reward function $r = \mathcal{R}(s, a)$. We use e to denote the robot embodiment and s_t, a_t, and r_t for state, action, and reward at timestep t, respectively. Within each episode, we sample an embodiment e from a distribution $\rho(e)$. Given a policy $\pi(\cdot \mid s, e)$, a trajectory can be generated by interacting with the environment, which is made up of a sampled embodiment and a sequence of states, actions, and rewards: $\tau = (e, s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_T, a_T, r_T)$, where T is the episode length. Similar to the setting of standard MDP, our goal is to learn a policy that maximizes the expected return across different embodiment-aware MDPs:

$$\pi^* := \arg\max_{\pi} \mathbb{E}_{\rho, \pi, P_E}\left[\sum_{t=1}^{T} r_t\right]. \quad (2)$$

We set the reward discount factor as 1 here since we assume each step is equally important.

B. Embodiment-aware Transformer

Trajectory representation. Different from Decision Transformer [37], we expect the trajectory representation to enable the Transformer to learn meaningful patterns between robot embodiment and actions, and the Transformer to conditionally generate actions based on embodiment at test time. Therefore, instead of feeding the returns-to-go as in [37],

Algorithm 1 Embodiment-aware Transformer (EAT)

Input: $D_{\text{expert}}^0, D_{\text{expert}}^1, \ldots, D_{\text{expert}}^{M-1}$
Output: Trained EAT Model
1: $D_{\text{expert}} = D_{\text{expert}}^0 \cup D_{\text{expert}}^1 \cup \cdots \cup D_{\text{expert}}^{M-1}$
2: **for** $i = 0, \ldots, I - 1$ iterations **do**
3:      Sample H-long (e, s, a, t) in D_{expert}
4:      Stack embeddings of (e, s, a) for each timestep
5:      Feed the Stacks to the GPT model
6:      Update the GPT model
7: **end for**

Algorithm 2 Real-world Evolution

Input: Trained EAT Model π_{EAT}, m_0
Output: $e^\star$
1: **for** $g = 0, \ldots, G - 1$ generations **do**
2:      **for** $t = 0, \ldots, T - 1$ timesteps **do**
3:          Action = $\pi_{\text{EAT}}(e_g, s_t, a_t, t)$
4:      **end for**
5:      Optimizer give the next candidate m_g
6: **end for**
7: **return** The best morphology $e^\star$.

---

we feed the vector representation e of the robot embodiment (e.g., length of legs and torso of the robot) at each timestep. This leads to the following trajectory representation which enables autoregressive training and generation:

$$\tau = (e, s_1, a_1, e, s_2, a_2, \ldots, e, s_T, a_T). \quad (3)$$

Note that although theoretically, the robot embodiment e in Eq. (3) can be time-varying, it is set as fixed during an episode in our experiment since varying the body shape of the real robot within an episode is impractical so far.

Training. For robots with M types of sampled embodiment, we are given a dataset D_{expert}^i of expert demonstration of offline trajectories τ for each of the i-th embodiment. We integrate these datasets and feed the sampled H timesteps of each trajectory into EAT. For token embeddings, we learn linear embedding layers for robot embodiment e, state s, and action a, with layer normalization [52]. Similar to [37], an embedding for each timestep is added to each token. Here, such a tuple (e, s, a) plays a similar role as a "word" in a language model. The tokens are then processed by a Generative Pre-trained Transformer (GPT) [53] model, which autoregressively predicts future tokens tuples (e, s, a). The predicted action is used for calculating the mean-squared error for backpropagation.

Evaluation. For evaluation, we can specify the current robot embodiment e and the robot's initial state, as the conditioning information to initiate generation. We feed the last H timesteps of the current trajectory τ into EAT to obtain the predicted action for the last observed state.

C. Evolution of Morphology with EAT

Once we have a unified control policy π_{EAT} for robots with varying morphology, we can further apply EAT to a real-world evolution task to find the best morphology:

$$e^\star = \underset{e \in \mathcal{E}}{\arg\max} f(e | \pi_{\text{EAT}}), \quad (4)$$

where the objective $f : \mathcal{E} \to \mathbb{R}$ is a black-box function that can be different from the reward function r. We can use an off-the-shelf black-box optimizer, such as BO, to solve this problem, as shown in Algorithm 2.

In this way, we solve the inefficiency challenge of the morphology-controller co-optimization problem by taking advantage of our embodiment-aware control mechanism.

IV. EVALUATIONS IN SIMULATION

Fig. 3. **The robot with changeable morphology.** We conduct the real-world experiment on a quadruped robot with variable lengths of lower limbs and torso. We show the robot with embodiment representation e of (0.27 m, 0.2 m, 0.2 m) and (0.35 m, 0.3 m, 0.3 m) on the top and the corresponding simulated counterparts on the bottom.

In this section, we focus on the locomotion training tasks of the quadruped robot Mini Cheetah [42] with variable lengths of the tibia and torso in simulation (Fig. 3). We want to train the robot to walk stably from scratch using EAT as described in Section III. The setup of this task is similar to [54]. Specifically, the state s here consists of: base linear and angular velocities, the gravity vector, joint positions and velocities, and the previous actions performed by the policy. The action a is the desired joint positions of the motors, sent to a PD controller. The total reward is a weighted sum of nine terms, as in [55], including velocity tracking error, velocity penalty, action penalty, and a bonus of feet-in-the-air duration. We defer the reward details to [54].

The embodiment vector e here consists of three dimensions: front tibia length, hind tibia length, and torso length. We sample some robots with specific embodiment e_i: 0.2, 0.25, and 0.3 m for front tibia length; 0.2, 0.25, and 0.3 m for hind tibia length; 0.2, 0.3, and 0.4 m for torso length. We use Proximal Policy Optimization (PPO) to train these 27 ($3 \times 3 \times 3$) robots separately with domain randomization [56] in Isaac Gym [55]. For all embodiment, we keep the reward function and initial state consistent. We collect 1000 trajectories from each environment for the training of EAT, while each trajectory has a length of 1000 timesteps.

We compare EAT with Embodiment-aware Behaviour Cloning (EABC), Vanilla Transformer, and a single PPO policy. For EABC, we concatenate the embodiment vector with the robot observation; for Vanilla Transformer, we use the same model architecture as EAT but without the embodiment vector e in the trajectory representation. We use the same training dataset as for EAT for these two methods. We also evaluate a baseline method: run a single PPO

Fig. 4. **Performance of trained policies on robots with different embodiments for a locomotion task of walking on a flat plane.** Each block corresponds to a specific robot embodiment—encoded by the length of torso, front tibia, and hind tibia—whose color represents the accumulated reward within an episode using a single trained policy. The best score for each robot embodiment is highlighted in yellow. In most cases, EAT can have the best performance compared with other alternatives.

policy—that trained on the original shape of Mini Cheetah—directly on robots with variable embodiments. Experimental results are shown in Fig. 4, where each block corresponds to a specific robot embodiment and the color represents the accumulated reward within an episode using the same trained policy. For each robot, we evaluate the performance of 1000 agents in parallel for 3 trails and report the average results.

A single EAT policy can successfully control robots with different shapes to walk: Except in some cases where robots have extremely unbalanced shapes, EAT can achieve a return higher than 10 in 70 / 80 cases. For 57 / 80 of the robot embodiments, EAT can outperform all other methods (scores highlighted in yellow). Specifically, for body shapes that are not in the training dataset (body length of 0.25 m and 0.35 m; front tibia and hind tibia of 0.15 m), EAT can still have a good walking score. For example, for the zero-shot evaluation of embodiment with a torso length of 0.4 m (the first line of the return matrix), EAT achieves the best score among these methods in 11 / 16 cases. This shows the generalization ability of EAT for unseen embodiments. EAT outperforms all other methods on average (Tab. I).

To evaluate the robustness of EAT, we add noises to the evaluation environment, including random friction coefficient, base mass, and pushing. The amplitudes of these noises are twice that used in the training stage. Results are shown in Tab. I, where EAT demonstrates its superiority even in a noisy environment. We will show in our real-world experiment that Transformer-based controllers are more robust than PPO, especially after the sim-to-real transfer.

We hypothesize that the superiority of EAT comes from the context information of previous tokens and the capacity of the model for fitting our diverse training dataset that contains both varying trajectories and robot morphology. To investigate the importance of access to previous states and actions, we ablate on the context length H.

The performance of EAT degrades significantly when the

TABLE I

SCORES OF LOCOMOTION USING DIFFERENT METHODS.

Method	Average Score	Average Score in Noisy Env.
EAT (Ours)	**18.87 ± 0.52**	**18.61 ± 0.13**
EABC	3.8 ± 0.9	2.18 ± 1.32
Transformer	13.32 ± 1.37	13.49 ± 1.38
PPO	16.53 ± 0.57	14.26 ± 1.01
EAT with $H = 1$	15.06 ± 0.53	14.15 ± 0.32
EAT with LD	13.54 ± 0.16	13.64 ± 1.19
EAT with LD-5k	14.19 ± 0.55	14.07 ± 0.67
EAT with LD-10k	15.32 ± 1.18	15.35 ± 0.34

horizon length is 1 (Tab. I), indicating that past information is essential for this morphology-varying locomotion task.

We further investigate the impact of the training dataset on the performance of EAT. We consider another training dataset that contains only 8 robot shapes instead of 27: torso length is 0.2 m or 0.4 m; front and back tibia length is 0.2 m or 0.3 m. We refer to this dataset as LD (Less Diverse). We also test on training datasets with variants of LD: LD with 5k trajectories per robot morphology and a 10k version, denoted as LD-5k and LD-10k.

Results in Tab. I show that training datasets with fewer types of robots (dataset LD) yield degraded performance of EAT, even if the dataset is larger (dataset LD-5k and LD-10k). This suggests that EAT can leverage the embodiment information in the dataset and implicitly build state-return associations via the similarity of the query and key vectors. This makes it superior in embodiment-conditioned control.

V. REAL-WORLD EXPERIMENT

We directly deploy the EAT model that is trained in the simulator to the real robots without any fine-tuning. The representation of the state, action, and embodiment remains the same as in simulation. We change the originally fixed tibias of the robot Mini Cheetah to configurable modules that we can easily adjust the length by replacing the steel

Fig. 5. **Robots with different morphologies controlled by the same EAT policy.** We run a trained EAT policy on robots with embodiment (0.3 m, 0.2 m, 0.2 m), (0.3 m, 0.2 m, 0.25 m), and (0.3 m, 0.25 m, 0.2 m) in each row of the figure respectively. EAT can successfully control the robot to move forward in these three cases where the positions of CoM are different.

Fig. 6. **Walking trajectories of robots with three different embodiments.** Embodiment 1 represents a body shape of the original Mini Cheetah, where the front lower limbs and hind lower limbs have the same medium length (0.2 m). In this case, both EAT and Vanilla Transformer can enable the robot to move forward, while PPO leads the robot to unstable states because of the reality gap. Embodiments 2 and 3 represent robots with asymmetrical front and hind lower limbs, leading to forward and backward CoM respectively. In these two cases, only EAT can successfully control the robot.

tube, and design 3D-printed parts to lengthen the torso, as shown in Fig. 3, with an accuracy of 1 cm.

A. EAT for Walking on Plane

We start from the same scenario as in the simulators: walking on real-world flat terrain. We do experiments on three different embodiments: $(0.3m, 0.2m, 0.2m)$, $(0.3m, 0.2m, 0.25m)$, and $(0.3m, 0.25m, 0.2m)$, referred to Embodiment 1, 2, and 3. These embodiments for testing represent three cases: 1) robots with the shape of the original Mini Cheetah; 2) robots with an asymmetric shape and a forward Center of Mass (CoM); and 3) robots with a backward CoM. However, our model is not limited by only these three embodiments. Rather, we use them as representative examples for demonstration purposes, and we expect that EAT can generalize to other possible embodiments.

Fig. 5 snapshots of successful walking with these three embodiments using EAT and Fig. 6 shows the walking distance in the first 15 or 25 seconds.

Experimental results of Embodiment 1 reveal the sim-to-real transfer ability of different control methods. In this test, EAT and Vanilla Transformer successfully controls the robot to walk stably. Robots with PPO can walk for the beginning few steps but quickly goes into unstable states as we can see in the accompanying video and Fig. 6. This is likely because of the reality gap that comes from the mismatch of motor models, floor material, and control latency. These reality gaps

continuously feed out-of-distribution (OOD) observations—that do not conform to the underlying distribution of the training data—to the policy. Thanks to the generalization ability of Transformers when decoding and the stable nature of offline RL, both EAT and Vanilla Transformer can overcome this reality gap and perform as in the simulator.

For Embodiment 2, PPO still leads the robot to unstable states very quickly. Robots with Vanilla Transformer fall forward after the first few steps, because of the unstable CoM of the robot. EAT successfully finds a control strategy to balance the robot and walk forward.

The backward CoM of Embodiment 3 makes the task harder since this would prevent the friction force to pull the robot forward. In this case, PPO still quickly leads the robots to unstable states, and only EAT can control the robot to stably walk forward.

B. EAT for Real-world Evolution

In this section, we showcase the application of EAT in Evolutionary Robotics: a robot can solve an unseen task by updating the morphology alone, using a fixed control policy. Here, we consider a locomotion task of walking down a stair of 10 cm, using the EAT policy that is trained in the simulator based only on training data of walking on flat terrains.

We evolve the robot morphology by optimizing the embodiment vector e: length of the torso, front lower limb, and back lower limb. For each generation, we use BO to sample a morphology e_g from ranges of (0.27 m, 0.35 m), (0.15 m,

Fig. 7. **Snapshots of the robot waking down a stair after morphology evolution.** Our robot with the EAT controller successfully find a morphology that suits this example task of stepping down: It finds a morphology of shorter front tibias (0.19 m), longer back tibias (0.22 m), and a longer torso (0.35 m). EAT successfully control the robot with this morphology, and this morphology helps the robot to keep balance during stepping down.

0.25 m), and (0.15 m, 0.25 m) respectively. We feed e_g to the EAT policy and test the walking performance—the furthest distance $f(e_g)$ that the robot can walk across the step—directly in the real world. We use BO to optimize $f(e_g)$ as regard to e_g for a maximum of 20 iterations. The furthest distance $f(e_g)$ is measured from the step to where the robot falls over. We assume that the robot can continuously walk on a flat plane once it walks for a distance of 1.5 m.

Fig. 8 compares the training curves when using the three strongest methods—EAT, Vanilla Transformer, and PPO—as the backend controllers while the morphology of the robot is updating.

Fig. 8. **Training curves of real-world evolution.** We compare the training processes using the same online morphology optimizer but different backend controllers. Since both PPO and Vanilla Transformer are sensitive to changes in robot morphology, only EAT can succeed in this task: The robot finds a morphology that can adapt to a terrain that is unseen beforehand.

EAT together with BO can successfully find the morphology that is capable of solving this task after around 10 iterations: The robot torso is extended to 0.35 m for increasing stability; the length of front tibias is lengthened at 0.22 m and that of hind tibias is shortened at 0.19 m, to provide a backward-oriented CoM that may help to keep balance while walking downstairs (Fig. 7).

Consistent with the previous experimental results, as PPO and Vanilla Transformer are sensitive to the morphology changes of our robots, they struggle in this real-world evolution task. Since it is hard for these methods to control a robot with variable morphology using a single policy, they keep falling when a new body shape is given, and hence keep giving the morphology optimizer noisy fitting values.

VI. DISCUSSION

Eiben [6] summarizes two limitations of the current state of the art in Evolutionary Robotics: 1) most studies rely on simulation for reproduction and evolution; 2) robot designs are usually very simple and driven by elementary open-ended control mechanisms. In this work, we attempt to solve these challenges by bridging the fields of big sequential data modeling [36] and evolvable hardware [57] for the first time.

Our proposed evolution pipeline is validated on a task of transversing stairs. Although some previous works have shown the ability of legged robots to climb stairs [58], [27], [59], our method goes in a different direction toward embodied intelligence: Unlike most of the previous work, our robot does not see any rough terrain in the simulation training phase, but use only online morphology optimization to overcome the unseen tasks. This unlocks the full potential for evolutionary robots to adapt to an unknown environment that is hard to predict or model beforehand.

Last, although the embodiment representation in our experiment involves only a three-dimension vector, a more sophisticated representation of robot shape—for example, pixel-based representation, which has been successfully embedded in a Transformer network in previous work [37], [47], [48]—can also be used in our framework. This inspires more real-world evolution applications involving more diverse morphologies in the future.

VII. CONCLUSIONS

In this work, we propose Embodiment-aware Transformer (EAT), a learning-based control method that takes actions conditioned on both state and embodiment. We pose this challenge as a conditional sequence modeling problem, and use Transformer to fit a data sequence of (embodiment, state, action) tuples. We train this model using trajectory data generated by multiple PPO policies from various morphologies. Evaluation results on a quadruped robot show that EAT outperforms all other alternatives in cross-embodiment tasks. By leveraging this feature, we apply EAT to a real-world evolution task, i.e., the robot updates its morphology alone online in the real world, for adapting to a locomotion task where the terrain is unseen beforehand. The success of EAT in solving this task shows the potential of embodiment-aware controller and Transformer in the application of Evolutionary Robotics. We hope that EAT can inspire a new push toward real-world robot evolution based on the recent advance in deep learning-based control.

REFERENCES

[1] R. A. Raff, *The shape of life: genes, development, and the evolution of animal form*. University of Chicago Press, 2012.

[2] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 1593–1599, IEEE, 2022.

[3] C. Yu, J. Cao, and A. Rosendo, "Learning to climb: Constrained contextual bayesian optimisation on a multi-modal legged robot," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9881–9888, 2022.

[4] S. Ha, P. Xu, Z. Tan, S. Levine, and J. Tan, "Learning to walk in the real world with minimal human effort," *arXiv preprint arXiv:2002.08550*, 2020.

[5] E. Massi, L. Vannucci, U. Albanese, M. C. Capolei, A. Vandesompele, G. Urbain, A. M. Sabatini, J. Dambre, C. Laschi, S. Tolu, *et al.*, "Combining evolutionary and adaptive control strategies for quadruped robotic locomotion," *Frontiers in Neurorobotics*, vol. 13, p. 71, 2019.

[6] A. Eiben, "Real-world robot evolution: why would it (not) work?," *Frontiers in Robotics and AI*, p. 243, 2021.

[7] T. F. Nygaard, C. P. Martin, J. Torresen, K. Glette, and D. Howard, "Real-world embodied ai through a morphologically adaptive quadruped robot," *Nature Machine Intelligence*, vol. 3, no. 5, pp. 410–419, 2021.

[8] T. F. Nygaard, C. P. Martin, D. Howard, J. Torresen, and K. Glette, "Environmental adaptation of robot morphology and control through real-world evolution," *Evolutionary Computation*, vol. 29, no. 4, pp. 441–461, 2021.

[9] T. F. Nygaard, D. Howard, and K. Glette, "Real world morphological evolution is feasible," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 1392–1394, 2020.

[10] T. F. Nygaard, C. P. Martin, E. Samuelsen, J. Torresen, and K. Glette, "Real-world evolution adapts robot morphology and control to hardware limitations," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 125–132, 2018.

[11] A. Rosendo, M. Von Atzigen, and F. Iida, "The trade-off between morphology and control in the co-optimized design of robots," *PloS one*, vol. 12, no. 10, p. e0186107, 2017.

[12] V. Vujovic, A. Rosendo, L. Brodbeck, and F. Iida, "Evolutionary developmental robotics: Improving morphology and control of physical robots," *Artificial life*, vol. 23, no. 2, pp. 169–185, 2017.

[13] M. Joachimczak, R. Suzuki, and T. Arita, "Artificial metamorphosis: Evolutionary design of transforming, soft-bodied robots," *Artificial life*, vol. 22, no. 3, pp. 271–298, 2016.

[14] L. K. Le Goff, E. Buchanan, E. Hart, A. E. Eiben, W. Li, M. De Carlo, A. F. Winfield, M. F. Hale, R. Woolley, M. Angus, *et al.*, "Morpho-evolution with learning using a controller archive as an inheritance mechanism," *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[15] L. K. L. Goff and E. Hart, "On the challenges of jointly optimising robot morphology and control using a hierarchical optimisation scheme," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1498–1502, 2021.

[16] 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 (TOG)*, vol. 39, no. 6, pp. 1–16, 2020.

[17] K. S. Luck, H. B. Amor, and R. Calandra, "Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning," in *Conference on Robot Learning*, pp. 854–869, PMLR, 2020.

[18] M. Jelisavcic, K. Glette, E. Haasdijk, and A. Eiben, "Lamarckian evolution of simulated modular robots," *Frontiers in Robotics and AI*, vol. 6, p. 9, 2019.

[19] A. Gupta, S. Savarese, S. Ganguli, and L. Fei-Fei, "Embodied intelligence via learning and evolution," *Nature communications*, vol. 12, no. 1, pp. 1–12, 2021.

[20] G. Picardi, H. Hauser, C. Laschi, and M. Calisti, "Morphologically induced stability on an underwater legged robot with a deformable body," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 435–448, 2021.

[21] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1–9, IEEE, 2018.

[22] M. V. Minniti, R. Grandia, F. Farshidian, and M. Hutter, "Adaptive clf-mpc with application to quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 565–572, 2021.

[23] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4730–4737, IEEE, 2019.

[24] H.-W. Park, K. Sreenath, J. Hurst, and J. Grizzle, "System identification and modeling for mabel, a bipedal robot with a cable-differential-based compliant drivetrain," in *Dynamic Walking Conference (DW), MIT*, vol. 6, 2010.

[25] U. Nagarajan, A. Mampetta, G. A. Kantor, and R. L. Hollis, "State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot," in *2009 IEEE International Conference on Robotics and Automation*, pp. 998–1003, IEEE, 2009.

[26] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[27] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," 2021.

[28] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[29] T. Chen, A. Murali, and A. Gupta, "Hardware conditioned policies for multi-robot transfer learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

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

[31] W. Huang, I. Mordatch, and D. Pathak, "One policy to control them all: Shared modular policies for agent-agnostic control," in *International Conference on Machine Learning*, pp. 4455–4464, PMLR, 2020.

[32] D. Pathak, C. Lu, T. Darrell, P. Isola, and A. A. Efros, "Learning to control self-assembling morphologies: a study of generalization via modularity," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[33] T. Wang, R. Liao, J. Ba, and S. Fidler, "Nervenet: Learning structured policy with graph neural networks," in *International conference on learning representations*, 2018.

[34] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath, and S. Levine, "Genloco: Generalized locomotion controllers for quadrupedal robots," *arXiv preprint arXiv:2209.05309*, 2022.

[35] R. K. Srivastava, P. Shyam, F. Mutz, W. Jaśkowski, and J. Schmidhuber, "Training agents using upside-down reinforcement learning," *arXiv preprint arXiv:1912.02877*, 2019.

[36] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[37] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15084–15097, 2021.

[38] M. Reid, Y. Yamada, and S. S. Gu, "Can wikipedia help offline reinforcement learning?," *arXiv preprint arXiv:2201.12122*, 2022.

[39] Q. Zheng, A. Zhang, and A. Grover, "Online decision transformer," *arXiv preprint arXiv:2202.05607*, 2022.

[40] M. Janner, Q. Li, and S. Levine, "Offline reinforcement learning as one big sequence modeling problem," *Advances in neural information processing systems*, vol. 34, pp. 1273–1286, 2021.

[41] R. Reddy, "Speech understanding systems: A summary of results of the five-year research effort at carnegie mellon university," *Pittsburgh, Pa*, 1977.

[42] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*, pp. 6295–6301, IEEE, 2019.

[43] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[45] N. Kitaev and D. Klein, "Constituency parsing with a self-attentive encoder," *arXiv preprint arXiv:1805.01052*, 2018.

[46] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.

[47] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3202–3211, 2022.

[48] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8844–8854, 2022.

[49] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, 2005.

[50] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *International conference on machine learning*, pp. 2052–2062, PMLR, 2019.

[51] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy q-learning via bootstrapping error reduction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[52] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[53] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

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

[55] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[56] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[57] G. W. Greenwood and A. M. Tyrrell, *Introduction to evolvable hardware: a practical guide for designing self-adaptive systems*, vol. 5. John Wiley & Sons, 2006.

[58] H. Shi, B. Zhou, H. Zeng, F. Wang, Y. Dong, J. Li, K. Wang, H. Tian, and M. Q.-H. Meng, "Reinforcement learning with evolutionary trajectory generator: A general approach for quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3085–3092, 2022.

[59] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," *arXiv preprint arXiv:2105.08328*, 2021.