000
001SUBEQUIVARIANTMORPHOLOGY-BEHAVIORCO-002
003EVOLUTION IN 3D ENVIRONMENTS

Anonymous authors

Paper under double-blind review



Figure 1: The left illustrates uncoordinated morphology and control without subequivariance, resulting in task failure. The right demonstrates symmetry in morphology and control through subequivariant co-evolution, enabling successful goal completion.

ABSTRACT

The co-evolution of morphology and behavior in 3D space has garnered considerable interest in the field of embodied intelligence. While recent studies have highlighted the considerable benefits of geometric symmetry for tasks like learning to locomote, navigate, and explore in dynamic 3D environments, its role within co-evolution setup remains unexplored. Existing benchmarks encounter several key issues: 1) the task lacks consideration for spatial geometric information; 2) the method lacks geometric symmetry to deal with the complexities in 3D environments. In this work, we propose a novel setup, named Subequivariant Morphology-Behavior Co-Evolution in 3D Environments (3DS-MB), to address the identified limitations. To be specific, we propose EquiEvo, which injects geometric symmetry, i.e., subequivariance, to construct dynamic, learnable local reference frames, enabling the joint policy to generalize to diverse task spatial structures, thereby improving co-evolution efficiency. Then, we evaluate EquiEvo on the proposed environments, where our method consistently and significantly outperforms existing approaches in tasks such as locomotion navigation and adversarial scenarios. Extensive experiments underscore the importance of subequivariance for the co-evolution of morphology and behavior, effective morphology-task mapping and robust morphology-behavior mapping.

037 038

039

004

006

008 009 010

011

013

014

015

016

017 018

019

021

024

025

026

027

028

029

031

032

034

1 INTRODUCTION

040 The rich diversity of animal morphologies, shaped over millions of years in complex environments, 041 underscores the deep link between body form and intelligence, where well-adapted morphologies 042 enable agents to learn and perform complex tasks effectively (Gupta et al., 2021; Brooks, 1991). 043 Morphology evolution is fundamentally driven by environmental interactions and task demands, 044 as supported by evolutionary biology (Maynard-Smith, 1974; Gould, 2010). The co-evolution of morphology and behavior in 3D environments has gained significant interest (Sims, 1994; Dong et al., 2023; Chen et al., 2023b; Gupta et al., 2021), focusing on how agents' form and function 046 evolve together to improve adaptability and performance. Some works, such as (Gupta et al., 2021; 047 Dong et al., 2023), impose bilateral symmetry constraints directly on morphology, diverging from 048 evolutionary principles, as agents should evolve optimal morphologies through interaction with the environment, rather than relying on predefined constraints.

Alternatively, the geometric symmetry of environments and dynamic systems naturally exists, which
 motivates us to explore how leveraging such symmetry can accelerate the evolution of optimal morphologies through interaction with the environment. Recent studies highlight that leveraging geometric symmetry in tasks like locomotion, navigation, and exploration within dynamic 3D settings

can significantly improve the efficiency of behavioral evolution (Chen et al., 2023a; 2024). While recent co-evolution benchmarks (Yuan et al., 2022; Huang et al., 2024; Liu et al., 2022) have achieved remarkable progress (Liu et al., 2022), they often fail to fully exploit geometric symmetry within 3D spaces. These benchmarks face two main issues: 1) a focus on tasks with limited spatial geometric considerations, such as the "move forward" objective, which are simplified to fixed directional movements and do not require complex direction-aware strategies; 2) a lack of research exploring the use of geometric symmetry in co-evolution setups to handle the directional complexities of 3D environments effectively.

062 In this work, we address the limitations of current co-evolution benchmarks by focusing on tasks 063 that require rich 3D spatial understanding and incorporate dynamic interactions. To achieve this, we 064 propose a novel setup, named Subequivariant Morphology-Behavior Co-Evolution in 3D Environments (3DS-MB). Our setup extends two core tasks to enable co-evolution in complex 3D spatial 065 environments: navigation and sumo (Chen et al., 2023a; Huang et al., 2024). In the navigation 066 task, the agent moves towards randomly generated goals, requiring continuous adaptation to new 067 spatial contexts. In the sumo task, two agents compete in an arena, where the objective is to push 068 the opponent out. The constantly changing adversarial dynamics require agents to adapt to vary-069 ing attack and defense angles, demanding both strategic behavior and morphology adaptation to effectively leverage rich directional information. These tasks introduce variability in movement and 071 environmental interaction, challenging agents to develop robust, generalized policies that account for diverse geometric conditions. 073

To effectively handle the directional complexities of 3D environments, we incorporate subequivari-074 ant neural networks within our co-evolution framework. As illustrasted in Figure 1, we introduce 075 subequivariant graph neural networks (Chen et al., 2023a; Han et al., 2022a) to predict a dynamic, 076 adaptive Local Reference Frame (LRF) for the agent (Chen et al., 2024). By projecting directional 077 vectors from the global world frame into this LRF, we transform spatial representations into an in-078 variant form, preserving geometric symmetry and simplifying the search space. For instance, under 079 rotational symmetry, states and actions in any direction can be treated as equivalent, effectively reducing a 2D/3D problem to a simpler 1D/2D one. This approach enables the policy to generalize 081 across diverse spatial structures, improving both generalization ability and sample efficiency in the 082 co-evolution process, allowing agents to efficiently learn and adapt to the complex directional in-083 formation inherent in 3D tasks. An essential aspect of our approach is ensuring that the evolution of morphology remains invariant to geometric transformations, such as rotations or translations, as 084 these do not alter the intrinsic properties of the environment. 085

Our contributions can be summarized as follows:

- We introduce 3DS-MB, a benchmark for subequivariant morphology-behavior coevolution, designed to capture the rich spatial geometric information and variability inherent in complex 3D environments.
 - We propose EquiEvo, a novel co-evolution framework that leverages subequivariant graph neural networks to effectively incorporate geometric symmetry, enhancing both morphology adaptation and behavior control.
 - We empirically validate the performance of EquiEvo on the 3DS-MB, where our approach consistently outperforms existing methods. Extensive ablation studies and analyses further demonstrate the framework's robustness and highlight the benefits of integrating subequivariance in co-evolution tasks.
- 098 099 100

087

090

092

093

094

095

096

2 PRELIMINARIES

101 102

Morphology-Behavior Co-Evolution. To account for the effect of morphology to agent's behavior, the typical MDP formulation could be extended by incorporating the agent's morphology. Specifically, we define the extended MDP as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma, \mathcal{G}_m)$ of state space, action space, transition dynamics, a reward function, a discount factor and the agent's morphology. \mathcal{G}_m is typically represented as a graph. The transition dynamics $P(s_{t+1}|s_t, a_t, \mathcal{G}_m)$ and the control policy $\pi(a_t|s_t, \mathcal{G}_m)$ are all conditioned on \mathcal{G}_m . The expected return depends on both the policy π and the morphology $\mathcal{G}_m: J(\pi, \mathcal{G}_m) \coloneqq \mathbb{E}_{\pi, \mathcal{G}_m} [\sum_{t=0}^{\infty} \gamma^t r_t]$. The morphology-behavior co-evolution problem ¹⁰⁸ can be mathematically described as optimizing the following two-layer objective:

$$\begin{aligned}
\mathcal{G}_{\mathrm{m}}^{*} &\coloneqq \operatorname*{arg\,max}_{\mathcal{G}_{\mathrm{m}}} J(\pi_{\mathcal{G}}, \mathcal{G}_{\mathrm{m}}) \\
\text{s.t.} \quad \pi_{\mathcal{G}} &= \operatorname*{arg\,max}_{\pi} J(\pi, \mathcal{G}_{\mathrm{m}}).
\end{aligned}$$
(1)

111 112 113

110

Building on the classical policy optimization algorithm, PPO (Schulman et al., 2017), Yuan et al. (2022) propose a unified approach to jointly optimize two sub-policies: morphology transformation and behavior control. Specifically, the policy is structured to first transform the agent's morphology and develop its attributes, which is then followed by behavior control for the resulting morphology within each episode.

Geometric Symmetry. Learning in 3D environments is hard due to the excess requirement of exploration. To relieve the difficulty of co-evolution in 3D environments with the massive search space, the inherent geometric symmetry of 3D environments should be exploited. Intuitively, if the entire environment undergoes rigid transformations such as rotations and translations, the perceived environment should remain unchanged from the perspective of an agent within it. In other words, the agent should remain unaffected by global geometric transformations. This property is formally captured by the concept of *equivariance*, defined as follows:

Definition 1 (Equivariance). Suppose \vec{z}^1 to be vector features (positions, velocities, etc) that are steerable by a group G, and **h** non-steerable scale features. A function f is G-equivariant, if for any transformation $g \in G, \forall \vec{z} \in \mathbb{R}^{3 \times m}, \mathbf{h} \in \mathbb{R}^d, f(g \cdot \vec{z}, \mathbf{h}) = g \cdot f(\vec{z}, \mathbf{h})$. Similarly, f is invariant if $f(g \cdot \vec{z}, \mathbf{h}) = f(\vec{z}, \mathbf{h})$. Here \cdot denotes the group operation.

130 To adhere to the principles of classical physics under the influence of gravity in typical 3D environ-131 ments, we consider equivariance over a specific subgroup of $E(3)^2$ defined as $E_{\vec{a}}(3)$. This subgroup is composed of $O_{\vec{q}}(3) \coloneqq \{ \boldsymbol{O} \in \mathbb{R}^{3 \times 3} \mid \boldsymbol{O}^\top \boldsymbol{O} = \boldsymbol{I}, \, \boldsymbol{O} \vec{\boldsymbol{g}} = \vec{\boldsymbol{g}} \} \text{ and } T_{\vec{\boldsymbol{g}}}(3) \coloneqq \{ \vec{\boldsymbol{t}} \in \mathbb{R}^3 \mid \vec{\boldsymbol{t}} \times \vec{\boldsymbol{g}} = \vec{\boldsymbol{0}} \}.$ 132 133 The group operation of $E_{\vec{q}}(3)$ is instantiated as $g \cdot \vec{z} \coloneqq O\vec{z} + \vec{t}^3$. In this way, $E_{\vec{q}}(3)$ -equivariance 134 is constrained to translations, rotations, and reflections along the direction of \vec{q} . We refer to this 135 property as *subequivariance*, specifically highlighting $E_{\vec{a}}(3)$ -equivariance. This relaxation of group 136 constraints is critical in environments influenced by gravity, as it allows the model to capture gravita-137 tional effects. Unless otherwise specified, in the following text, we will use the term "equivariance" to refer to $E_{\vec{q}}(3)$ -equivariance, and "invariance" to refer to $E_{\vec{q}}(3)$ -invariance. 138

$$f_{\vec{g}}(\vec{z}, h) = [\vec{z}, \vec{g}] \sigma([\vec{z}, \vec{g}]^\top [\vec{z}, \vec{g}], h),$$
(2)

147 148 149

150

151 152

153

154

156

157 158

159

where $\sigma(\cdot)$ is an Multi-Layer Perceptron and [] denotes concatenation along the last dimension.

3 Setup and method

Our goal is to learn a co-evolution policy in 3D environments that simultaneously adapts the agent's morphology and enhances behavior control skills to optimize task performance. To achieve this, we first introduce our newly proposed 3D morphology-behavior co-evolution setup, termed 3DS-MB,

Table 1:	Comparison	of Setups.
----------	------------	------------

Aspect	Prior	3DS-MB
Task Graph Group	Fixed Directions Topology Graph	Variable Directions Subequivariant Graph
Oroup	V	$L_{\tilde{g}}(3)$

¹Note that we use a right-arrow superscript on \vec{z} to distinguish it from the scalar h, which remains unaffected by the transformations.

²The symmetrical structure of 3D environments is captured by E(3), the 3-dimensional Euclidean group, which includes rotations, reflections, and translations.

³Note that \vec{t} only acts on the 3D coordinate vector.

which incorporates tasks involving richer 3D information. We then present our co-evolution frame work, EquiEvo (illustrated in Figure 2), which leverages the symmetric structure of 3D environ ments through subequivariant neural networks to efficiently evolve morphology and learn control
 skills. We also outline our method in Algorithm 1.

1663.1 EXTENSION TO 3DS-MB

Setups. Exploring how agents evolve and behave in tasks involving complex state information of 3D environments is crucial, as factors like positions, velocities, and directions are inherently represented as vectors.

171 While prior works have made significant strides, 172 existing benchmarks (Yuan et al., 2022; Huang 173 et al., 2024) primarily focus on tasks with fixed 174 directions and limited consideration of those re-175 quiring richer vector representations, particu-176 larly tasks involving variable directions. Consequently, there is no need to utilize symme-177 try groups, and modeling with topology graphs 178 is sufficient for co-evolution of morphology and 179 behavior in their tasks. In contrast, our pro-180 posed 3DS-MB setup introduces tasks with vari-181 able directions, employs subequivariant graphs, 182 and leverages the group $E_{\vec{q}}(3)$ to capture the ge-183 ometric properties of 3D spaces. A detailed com-184 parison of the environment setups between 3DS-185 MB and prior benchmarks is provided in Table 1.

186

187 Invariant of Morphology Value. The evolu-188 tion of morphology should be invariant to geo-189 metric transformations like rotations or translations, as these do not alter the intrinsic proper-190 191 ties of the environment. From the perspective of PPO optimization, to optimize the morphology 192 transform policy, we need to estimate the value 193 of morphology transform actions, which must be 194 invariant to $E_{\vec{q}}(3)$ transformations. For exam-195 ple, consider a scenario where an agent with the 196 same morphology is initialized in a state that has 197 been rotated within the environment. The behavior should rotate correspondingly, ensuring that 199 the morphology value determined by behavior re-200 mains consistent. In co-evolution frameworks 201 like Transform2Act (Yuan et al., 2022), morphology transformation and behavior control are 202 treated as different steps within a single episode. 203



Figure 2: Flowchart of EquiEvo.

Consistent feedback for morphology transformation can only be achieved by leveraging geometric
 symmetry throughout the interaction with the environment. To achieve this, we integrate equivariant
 networks into the behavior control policy. By inject symmetry into the model architecture, the output is constrained to be equivariant, provides consistent value for morphology value estimation, and
 facilitates the co-evolution learning process. Furthermore, equivariant networks have been proven
 to learn control policies more efficiently, which is crucial for reducing the complexity of solving the
 complex two-layer optimization problem. The role of geometric symmetry in morphology evolution
 is discussed in detail in Appendix D.

211 212

213

3.2 EquiEvo

Agent Representation. The morphology of the agent is a graph $\mathcal{G}_m := (\mathcal{V}, \mathcal{E}, \mathbf{H}^m)$. where each node $i \in \mathcal{V}$ represents a limb, and each edge $(i, j) \in \mathcal{E}$ corresponds to the joint that connects the limbs *i* and *j*. The morphology structure of the agent is represented by the topology of graph. \mathbf{H}^m

contains morphology attributes for the agent's limbs, joints, and the specific design for the structure given by the graph. For each node i, $h_i^m \in \mathbb{R}^{C_m}$ includes the length and size of the limb, joint torque limit, the node depth in the body tree, etc.

219 The agent's state is defined as a subequivariant graph $\vec{\mathcal{G}}_{\vec{g}} := (\mathcal{V}, \mathcal{E}, \boldsymbol{H}^{\mathrm{m}}, \boldsymbol{H}^{\mathrm{b}}, \vec{\boldsymbol{Z}}, \vec{\boldsymbol{g}}) =$ 220 $(\mathcal{G}, H^{\mathrm{b}}, \vec{Z}, \vec{g})$. Each node *i* has scalar state and vector state, expressed as $h_i^{\mathrm{b}} \in H^{\mathrm{b}}$ and $\vec{z}_i \in \vec{Z}$, 221 respectively. The scalar state $h_i^{\rm b} \in \mathbb{R}^{C_{\rm b}}$ contains scalars representing the configuration of the joint, 222 including joint angle, joint angular velocity, etc. The vector joint state $\vec{z}_i \in \mathbb{R}^{3 \times C_m}$ contains posi-223 tion $\vec{p}_i \in \mathbb{R}^3$, direction, linear velocity, angular velocity, etc. Here, position \vec{p}_i is transformed into a 224 translation-invariant representation by subtracts position of root node, thereby ensuring translation 225 invariance. Notably, the morphology attributes and the scalar state are invariant, while the vector 226 state is equivariant. 227

We emphasize the differences between \mathcal{G}_{m} and $\mathcal{G}_{\vec{g}}$. \mathcal{G}_{m} solely represents the agent's morphology and does not interact with the environment. In contrast, $\mathcal{G}_{\vec{g}}$ interacts with the environment, involving H^{b} and \vec{Z} , which are obtained during the rollout process.

232 **Morphology Transform.** In the morphology transform stage, starting from the original morphol-233 ogy, the agent undergoes morphology transform actions $a^{\rm m}$ to modify the structure and attributes through the morphology transform sub-policy π_{θ}^{m} . We adopted the two-stage design introduced in Transform2Act (Yuan et al., 2022). The transform actions $a^{m} \in \{a^{s}, a^{a}\}$ include two types: 234 235 (1) Structure transform actions a^{s} , which are discrete and change the topological structure of the 236 morphology by adding or removing joints and limbs; (2) Attribute transform actions $a^{\rm a}$, which 237 can be continuous or discrete, used to modify the specific instantiation of the given structure. Ac-238 cordingly, π_{θ}^{n} is a composition of the structure transform sub-policy π_{θ}^{s} and the attribute transform 239 policy π_{θ}^{a} . We divide the morphology transform stage into two sub-stages - structure transform stage 240 and attribute transform stage - and use Φ as the stage identifier. We can represent the morphology 241 transform sub-policy as follows: 242

$$\pi_{\theta}^{\mathrm{m}}(\boldsymbol{a}^{\mathrm{m}}|\mathcal{G}_{\mathrm{m}}, \Phi) \coloneqq \begin{cases} \pi_{\theta}^{\mathrm{s}}(\boldsymbol{a}^{\mathrm{s}}|\mathcal{G}_{\mathrm{m}}, \Phi), & \text{if } \Phi = \text{Structure} \\ \pi_{\theta}^{\mathrm{a}}(\boldsymbol{a}^{\mathrm{a}}|\mathcal{G}_{\mathrm{m}}, \Phi), & \text{if } \Phi = \text{Attribute} \end{cases}$$
(3)

244 245

251

252

253

254

255

256

257

258 259

243

231

The structural transform stage lasts for $N_{\rm s}$ steps, while the attribute transform stage spans $N_{\rm a}$ steps. The resulting morphology is then utilized by the behavior sub-policy to interact with the environment. Notably, during the morphology transform stage, no environment reward is assigned to the agent, as there is no direct interaction with the environment. Instead, the morphology transform sub-policy is updated solely based on feedback from the subsequent behavior control stage.

Local Reference Frame Canonicalization. According to Han et al. (2024), the Local Reference Frame (LRF) Canonicalization is a common technique in geometric graph neural networks to achieve equivariance (or invariance). We chose to use the LRF in our work for its simplicity and effectiveness. During the behavior control stage, at each step, we first predict a local reference frame (LRF) using subequivariant graph neural networks. We then perform canonicalization by projecting the vector states from the global world frame into the LRF as follows:

$$\mathcal{G}_{\vec{q}} \leftarrow \mathrm{LRF}(\vec{\mathcal{G}}_{\vec{q}}).$$
 (4)

To obtain LRF, We use subequivariant graph neural networks φ to process $\vec{\mathcal{G}}_{\vec{g}}$, predict two candidate vectors and perform orthonormalization OP:

$$\vec{u}, \vec{v} \leftarrow \varphi\left(\vec{\mathcal{G}}_{\vec{g}}\right),$$
(5)

$$\boldsymbol{O} \leftarrow \operatorname{OP}(\boldsymbol{\vec{u}}, \boldsymbol{\vec{v}}).$$
 (6)

266 Specifically, φ consists of *L* subequivariant message passing layers. The message passing mecha-267 nism operates as follows: we first construct edge features from the node features, and then use both 268 $\phi_{\vec{g}}$ and $\psi_{\vec{g}}$, as described in Equation (2), to generate messages. These messages are subsequently 269 aggregated and merged with the original state. For details on subequivariant message passing, please refer to Appendix C. After message passing, we predict \vec{u} and \vec{v} by using the output vector state $\vec{z}_1^{(L)}$ of the root node:

$$\vec{u} \leftarrow \vec{z}_1^{(L)} W_{\vec{u}},\tag{7}$$

272 273

281

283

284 285

288

298 299 300

301

302

303

304 305 306

307 308

309 310

311

$$\vec{v} \leftarrow \vec{z}_1^{(L)} W_{\vec{v}},$$
 (8)

where $W_{\vec{u}} \in \mathbb{R}^{m \times 1}$ and $W_{\vec{v}} \in \mathbb{R}^{m \times 1}$.

Now we use \vec{u} and \vec{v} to construct a orthonormal basis. As \vec{z} only consider $O_{\vec{g}}(3)$ -equivariant transformations, we could set one of the axes as parallel to \vec{g} . We have $O := [\vec{e}_1, \vec{e}_2, \vec{e}_3] \leftarrow OP(\vec{u}, \vec{v})$:

$$\vec{\boldsymbol{e}}_3 \leftarrow [0,0,1]^\top \tag{9}$$

$$\vec{e}_1 \leftarrow \frac{\vec{u} - \langle \vec{u}, \vec{e}_3 \rangle \vec{e}_3}{\|\vec{u} - \langle \vec{u}, \vec{e}_3 \rangle \vec{e}_3 \|},\tag{10}$$

$$\vec{e}_2 \leftarrow \frac{\vec{v} - \langle \vec{v}, \vec{e}_1 \rangle \vec{e}_1 - \langle \vec{v}, \vec{e}_3 \rangle \vec{e}_3}{\|\vec{v} - \langle \vec{v}, \vec{e}_1 \rangle \vec{e}_1 - \langle \vec{v}, \vec{e}_3 \rangle \vec{e}_3\|}.$$
(11)

Having obtained O, we project the agent's vector state into the local reference frame to obtain the invariant vector state:

$$\boldsymbol{z}_i \leftarrow \boldsymbol{O}^\top \boldsymbol{\bar{z}}_i^{(0)}, \tag{12}$$

where $\vec{z}_i^{(0)} \in \mathbb{R}^{3 \times m}$ is the input vector state of node i in the global world frame, and $z_i \in \mathbb{R}^{3 \times m}$ is the invariant vector state in the local reference frame.

Finally, we transform the subequivariant graph of the agent's state into an invariant graph $\mathcal{G}_{\vec{g}} := (\mathcal{G}_{\mathrm{m}}, H^{\mathrm{b}}, Z, \vec{g}) \leftarrow \mathrm{LRF}(\vec{\mathcal{G}}_{\vec{g}})$. A formal proof of the invariance of LRF Canonicalization is presented in Theorem 1.

Behavior Control. We use the invariant graph $\mathcal{G}_{\vec{g}}$ (such as GraphConv (Kipf & Welling, 2017)) to perform behavior control. We use a conventional neural network $\varphi_{\theta}^{\rm b}$ (such as MLP) to instantiate the behavior control sub-policy $\pi_{\theta}^{\rm b}$ and predict the behavior control action $a^{\rm b}$:

$$a^{\mathrm{b}} \leftarrow \varphi^{\mathrm{b}}_{\theta} \left(\mathcal{G}_{\vec{g}} \right).$$
 (13)

Value Estimation. In the value estimation process, for greater flexibility, we employ separate individual models with the same architecture as φ in Equation (5) to predict $\mathcal{G}_{\vec{g}}^V$. The invariant graph $\mathcal{G}_{\vec{g}}^V$ is then used for value estimation. We instantiate the value function V_{ϕ} using a conventional neural network φ_{ϕ} (such as MLP), which predicts the value V as follows:

$$V \leftarrow \varphi_{\phi}\left(\mathcal{G}_{\vec{q}}^{V}\right).$$
 (14)

Formal proof of the invariance of the co-evolution are presented in Appendix A.

4 EXPERIMENTS

4.1 3DS-MB Environments

312 To evaluate our EquiEvo method, we extend 313 the morphology-behavior co-evolution envi-314 ronments (Yuan et al., 2022; Huang et al., 315 2024), based on the MuJoCo simulation en-316 gine (Todorov et al., 2012), to operate within 317 a practical three-dimensional physical environ-318 ment. We refer to this new environment as 319 3DS-MB. In the 3DS-MB environments, the agent's morphology design and state are rep-320 resented by the morphology graph \mathcal{G}^{m} and the 321 subequivariant graph $\vec{\mathcal{G}}_{\vec{q}}$. We apply the 3DS-322 MB environments to three different tasks, as 323 shown in Figure 3.



Figure 3: Illustrations of Tasks. Navigation: the agent navigates towards the target (green). Sumo: Two agents compete against each other to push the opponent out of the arena.



Humanoid Navigation. 1. Initial Conditions. The agent's position and orientation are initialized 338 randomly, with the goal point placed randomly within a radius of [3, 4] from the agent. Each time 339 the agent reaches the goal, a new goal point is generated randomly, requiring the agent to continually 340 adjust its direction of movement to reach as many goal points as possible. For this task, we use a humanoid morphology. To maintain the humanoid structure, we skip the structural transform stage 342 and start directly with the attribute transform. 2. Termination. The state must simultaneously 343 satisfy isfinite(\vec{p}), isfinite(\vec{v}), and $h \in [minHeight, maxHeight]$, where \vec{p} , \vec{v} , and h represent the position, velocity, and height of the agent, respectively. The values minHeight and maxHeight are 344 the preset minimum and maximum heights. If any of these conditions are not met, the training is terminated. For the humanoid agent, we set $h \in [1,2]$. **3. Reward.** For the navigation task, we 346 design a reward structure for the humanoid agent that consists of six components: success bonus, forward reward, distance reward, control cost, contact cost, and survive reward. The total reward is 348 the sum of these components. For more details, please refer to Appendix E.1. 349

350

336 337

341

345

347

351 352

353

354

355

356

357

Ant Navigation. 1. Initial Conditions. The initial setup is similar to that of the Humanoid Navigation task. In this task environment, we use the ant morphology for experimentation, incorporating the structure transform stage, attribute transform stage, and behavior control stage during the evolution process. 2. Termination. The termination conditions mirror those of the Humanoid Navigation task. For the ant agent, we set $h \in [0.28, 0.8]$. **3. Reward.** For the navigation task, we design a reward structure for the Ant agent consisting of four components: success bonus, distance reward, control cost, and survive reward. The total reward is the sum of these components. For further details, please refer to Appendix E.1.

358 359 360

361 **Ants Sumo.** 1. Initial Conditions. The positions and orientations of two agents are randomly 362 initialized, requiring them to learn to attack and defend in various strategic situations while attempt-363 ing to push each other out of the arena. The radius of the arena is randomized within the range of 364 [2.5, 4.5], and the arena has a fixed height of 0.5. Additionally, the agents are permitted to reach a maximum height of 0.29 above the arena ground. In this environment, we use the ant morphology 366 for experimentation. Following the CompetEvo (Huang et al., 2024) setup, the structure transform 367 stage is omitted, retaining only the attribute transform stage and behavior control stage. 2. Termi-368 nation. The objective is for one agent to win by disqualifying the opponent, which occurs when an agent pushes the other out of the arena or when an agent's height exceeds the set limit (draw 369 outcomes). The disqualified agent loses, ending the episode. 3. Reward. For the Ants Sumo task, 370 we design a reward structure consisting of seven components: win bonus, lose bonus, draw bonus, 371 control cost, push opponent reward, move reward, and survive reward. The total reward is the sum 372 of these components. For more details, please refer to Appendix E.1. 373

374 We chose to evolve morphology skeletons from scratch to address the most challenging setup and 375 eliminate biases from prior knowledge, starting the Ant Navigation task with a simple "atomic morphology" (a torso body). For other starting morphologies, such as in the Humanoid Navigation 376 and Ants Sumo tasks, we focused solely on attribute transformations to maintain task-specific con-377 straints, as structural transformation boundaries are less well-defined.



Figure 5: Training and Evaluation Curves for Sumo.

4.2 BASELINES, METRICS, AND IMPLEMENTATIONS

Baselines. For Agent X(Ant/Humanoid), we employ the EquiEvo method to obtain EquiEvoX. To conduct a comprehensive comparison, we define the other three baseline methods: EquiX, EvoX, X. EquiX maintains subequivariance but removes the morphology transform component. EvoX includes morphology transform but does not integrate subequivariance. X is the most baseline method, lacking both subequivariance and morphology transform component. For the three types of tasks, the specific methods adopted are as follows:

400 401

392 393

394 395

396

397

398

399

- 402 403
- Ant Navigation: EquiEvoAnt, EquiAnt, EvoAnt, Ant.Humanoid Navigation: EquiEvoHumanoid, EquiHumanoid, EvoHumanoid, Humanoid.
- 404 405
- Ants Sumo: EquiEvoAnt, EquiAnt, EvoAnt, Ant.

The Ant Navigation uses the Transform2Act (Yuan et al., 2022) codebase as the baseline, while
the Humanoid Navigation and Ant Sumo use the CompetEvo (Huang et al., 2024) codebase as the
baseline. For more information on the baselines, please refer to Appendix E.2.

Metrics. For Navigation task, we use cumulative reward to assess the performance of the agents; for Sumo task, we use win rate (*Win Rate = #win_episode / #all_episode*). Each experiment is trained with 3 seeds to report the average and standard deviation of the cumulative reward or win rate.

412 **Implementations.** We use PPO (Schulman et al., 2017) as the reinforcement learning algorithm 413 in all experiments. We implemented EquiEvo based on the Transform2Act codebase (Yuan et al., 414 2022) and the CompetEvo codebase (Huang et al., 2024), which are built on the PyTorch framework. 415 There is no weight sharing between the policy network π_{θ} and the value estimation network V_{ϕ} . The 416 value of the maximum timesteps of an episode is 1,000 (Navigation) / 500 (Sumo). The value of the 417 maximum epoch is 500 (Ant) / 2,000 (Humanoid). Additionally, in Sumo, we adopt Bansal et al. 418 (2018)'s approach, where teams using different methods compete against each other within an arena. 419 For detailed hyperparameters, refer to Table 2.

- 420 421
- 421 4.3 MAIN RESULTS 422

Navigation. We first conduct a performance comparison of methods on the navigation task, as shown in Figure 4.

For the Humanoid agent, *EquiEvo* surpasses baseline
methods across random scenarios (Figure 4b). The ranking from best to worst—*EquiEvoHumanoid*, *EquiHu- manoid*, *Humanoid*, and *EvoHumanoid*—highlights the
significance of subequivariance in morphology-behavior
co-evolution. Without subequivariance, applying morphology transform alone (*Evo*) expands the search space,



Figure 6: Visualization of Humanoid Morphology.



Figure 7: Comparisons with Hand-Craft Normalization.

hindering efficient training. This demonstrates the crucial role of subequivariance in optimizing co-evolution.

The visualization in Figure 6 demonstrates that subequivariance significantly impacts the shape of
the humanoid agent. *EvoHumanoid*, developed without geometric symmetry constraints, lacks balance, resulting in poor coordination and stability during movement. In contrast, *EquiEvoHumanoid*develops a more coordinated morphology, leading to greater stability and flexibility—more akin to
that of a professional athlete.

For Ant, the experimental results (Figure 4a) rank *EquiEvoAnt* as the best in locomotion navigation,
followed by *EvoAnt*, *EquiAnt*, and *Ant*. The *EquiEvo* method enhances learning by integrating
subequivariance and morphology transform, demonstrating the effectiveness of both components in
our approach. *EvoAnt* performs well due to its morphology transform, while *EquiAnt* is weaker
without it but still outperforms *Ant*.

458 459 460

461

462

463

464

444 445

446

447

Sumo. In the sumo task (Figure 5), the *EquiEvoAnt* agent consistently outperformed all baselines. The results emphasize the critical impact of subequivariance, which proved more influential than morphology evolution alone in enhancing performance. However, integrating morphology evolution with subequivariance yielded a synergistic effect, further boosting the agents' capabilities. This highlights that subequivariance is key to success in sumo tasks, with the addition of morphology evolution refining and optimizing strategies for significant performance gains.

465 466 467

468

4.4 ABLATIONS AND ANALYSES

Comparisons with Hand-Crafted Normalization. In addition to our equivariant network approach, hand-crafted methods can also construct a Local Reference Frame (LRF). To compare the effectiveness of these approaches, we designed the following experiments, ablated in Figure 7: 1. **Evo+DN**: a variant using hand-crafted normalization where \vec{z} are treated as scalars and the goal direction is used to construct the LRF; **2. Evo+HN**: another hand-crafted normalization variant, treating \vec{z} as scalars but constructing the LRF based on the agent's heading direction; **3. Evo**: a non-equivariant variant that treats \vec{z} as scalars without any LRF construction.

Can equivariant network methods replace or even surpass hand-crafted normalization? Experiments demonstrate that regardless of using goal or heading direction to construct the LRF, hand-crafted methods perform worse than ours, indicating that equivariant networks are more effective for morphology evolution, offering a plug-in solution for equivariant adaptation.

Do different tasks require different LRF designs for symmetry? We observe that EvoAnt, a goal-directed task, benefits more from DN since the primary reward comes from approaching the goal.
 In contrast, the humanoid task, which requires both goal-reaching and forward motion, sees better performance with HN. This suggests that optimal LRF design varies by task. Additionally, evidences from prior works (Chen et al., 2023a; 2024) indicate that learning-based equivariant methods, which automatically identify the optimal LRF by leveraging geometric information, outperform hand-crafted approaches that rely on manually applying symmetry priors.



Figure 8: Ablations of Equivariance on EvoAnt.

Figure 9: Analyses of Morphology-Task.

Ablations of Equivariance. In EquiEvo (or EquiActorCritic), both the Actor and Critic networks are designed to be subequivariant, ensuring that the entire reinforcement learning process adheres to equivariance under the Actor-Critic framework. To evaluate the impact of each network, we perform ablation experiments where one network retains subequivariance while the other does not, producing the variants EquiActor and EquiCritic, respectively. The results in Figure 8 demonstrate that the performance of EquiEvo diminishes when either the Actor or Critic lacks subequivariance, emphasizing the necessity of this property in both networks. Moreover, the Actor's subequivariance appears to play a more crucial role than the Critic's, likely because of its direct impact on decision-making and action selection. These findings underscore the importance of maintaining subequivariance in both the Actor and Critic to achieve an efficient and consistent reinforcement learning process.

506 507

496

497

498

499

500

501

502

503

504

505

Analyses of Morphology-508 Mapping. The Task 509 evolved morphology of 510 *EquiEvoAnt* exhibits strong 511 symmetry, as seen in Fig-512 ure 10, which coincides 513 with the symmetric struc-514 ture of the task reward. 515 To investigate how task 516 design affects morphology 517 evolution, we introduce a 518 forward reward to EvoAnt and train both EquiEvoAnt 519



Figure 10: Step-by-step visualization of the morphology transformation within a single episode, corresponding to the final checkpoint . (a) EvoAnt; (b) EquiEvoAnt; (c) EquiEvoAnt with Forward Reward.

and EvoAnt under this modified reward structure. The reward curves, presented in Figure 9, indicate 520 that EquiEvoAnt continues to outperform EvoAnt in this altered task. In Figure 10 (b) and (c), 521 we compare the evolution processes under different task (reward) settings. For the task with a 522 forward reward, the evolved morphology exhibits a laterally symmetric structure with stronger 523 front legs and weaker hind legs, while the task without a forward reward leads to an radially 524 symmetric morphology. This shows that the evolution of morphology is fundamentally shaped by 525 environmental interactions and task demands, rather than being a predefined goal. The comparison 526 between Figure 10 (a) and (b) shows that methods without integrating geometric equivariance, 527 due to lower sample efficiency, fail to fully evolve a radially symmetric morphology that meets 528 task requirements. This highlights the necessity of considering geometric equivariance in the co-evolution of morphology and behavior frameworks. 529

530

533

Analysis of Morphology-Behavior Mapping. We conduct experiments on the Ant Navigation
 task to analyze robust Morphology-Behavior Mapping, with details in Appendix E.3.

534 5 CONCLUSIONS

We present EquiEvo, a 3D subequivariant framework for co-evolving morphology and behavior,
leveraging geometric symmetries to enhance efficiency and adaptability. Our results demonstrate its
effectiveness in navigation and sumo tasks, highlighting the importance of geometric equivariance
in driving task-specific morphologies through interaction-driven optimization. Further discussion
on structural evolution is provided in Appendix E.4. Future work can explore more complex environments, tasks, and evolution settings to further advance embodied intelligence and robotic design.

540 REFERENCES

572

- Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations*, 2018. 4.2
- 545 Cenk Baykal and Ron Alterovitz. Asymptotically optimal design of piecewise cylindrical robots
 546 using motion planning. In *Robotics: science and systems*, volume 2017. Orlando, FL, USA,
 547 2017. B
- Rodney A. Brooks. New approaches to robotics. *Science*, 253(5025):1227–1232, 1991. doi: 10. 1126/science.253.5025.1227. 1
- Runfa Chen, Jiaqi Han, Fuchun Sun, and Wenbing Huang. Subequivariant graph reinforcement learning in 3d environment. In *International Conference on Machine Learning*. PMLR, 2023a. 1, 2, 4.4, B
- Runfa Chen, ling Wang, Yu Du, Tianrui Xue, Fuchun Sun, Jianwei Zhang, and Wenbing Huang.
 Subequivariant reinforcement learning in 3d multi-entity physical environments. In *International Conference on Machine Learning*. PMLR, 2024. 1, 2, 4.4, B
- Tianjian Chen, Zhanpeng He, and Matei Ciocarlie. Hardware as policy: Mechanical and computational co-optimization using deep reinforcement learning. *ArXiv preprint*, abs/2008.04460, 2020. URL https://arxiv.org/abs/2008.04460. B
- Xinglin Chen, Da Huang, Minglong Li, Yishuai Cai, Zhuoer Wen, Zhongxuan Cai, and Wenjing
 Yang. Evolving physical instinct for morphology and control co-adaption. In 2023 IEEE/RSJ
 International Conference on Intelligent Robots and Systems (IROS), pp. 6616–6623. IEEE, 2023b.
 1, B
- Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution*, 7(1): 11–23, 2014. B
- Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016a. B
- ⁵⁷¹ Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016b. B
- Ruta Desai, Ye Yuan, and Stelian Coros. Computational abstractions for interactive design of robotic devices. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1196–1203. IEEE, 2017. B
- Heng Dong, Junyu Zhang, Tonghan Wang, and Chongjie Zhang. Symmetry-aware robot design with
 structured subgroups. In *International Conference on Machine Learning*, pp. 8334–8355. PMLR,
 2023. 1, E.4
- Stephen Jay Gould. *The panda's thumb: More reflections in natural history*. WW Norton & company, 2010. 1
- Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021. 1, B, E.4
- David Ha. Reinforcement learning for improving agent design. *Artificial life*, 25(4):352–365, 2019.
 B, E.4
- 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*, volume 13, pp. 10–15607, 2017. B
- Jiaqi Han, Wenbing Huang, Hengbo Ma, Jiachen Li, Josh Tenenbaum, and Chuang Gan.
 Learning physical dynamics with subequivariant graph neural networks. In Sanmi Koyejo,
 S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 December 9,

594 595 596	2022, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/a845fdc3f87751710218718adb634fe7-Abstract-Conference.html. 1,2
597 598	Jiaqi Han, Yu Rong, Tingyang Xu, and Wenbing Huang. Geometrically equivariant graph neural networks: A survey. <i>arXiv preprint arXiv:2202.07230</i> , 2022b. B
599 600 601 602	Jiaqi Han, Jiacheng Cen, Liming Wu, Zongzhao Li, Xiangzhe Kong, Rui Jiao, Ziyang Yu, Tingyang Xu, Fandi Wu, Zihe Wang, et al. A survey of geometric graph neural networks: Data structures, models and applications. <i>arXiv preprint arXiv:2403.00485</i> , 2024. 2, 3.2
603 604 605	Jiaheng Hu, Julian Whitman, and Howie Choset. Glso: grammar-guided latent space optimization for sample-efficient robot design automation. In <i>Conference on Robot Learning</i> , pp. 1321–1331. PMLR, 2023. B
606 607 608	Kangyao Huang, Di Guo, Xinyu Zhang, Xiangyang Ji, and Huaping Liu. Competevo: Towards morphological evolution from competition. <i>ArXiv preprint</i> , abs/2405.18300, 2024. URL https://arxiv.org/abs/2405.18300. 1, 3.1, 4.1, 4.1, 4.2, B, E.4
609 610 611 612 613	Donald Joseph Hejna III, Pieter Abbeel, and Lerrel Pinto. Task-agnostic morphology evolution. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id= CGQ6ENUMX6. B
614 615 616 617	Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional net- works. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. URL https: //openreview.net/forum?id=SJU4ayYgl. 3.2
618 619 620	Xingyu Liu, Deepak Pathak, and Kris M Kitani. Revolver: Continuous evolutionary models for robot-to-robot policy transfer. <i>arXiv preprint arXiv:2202.05244</i> , 2022. 1
621	John Maynard-Smith. Models in ecology. Cambridge university press, 1974. 1
622 623 624 625	Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to con- struct and control agents using deep reinforcement learning. In 2019 international conference on robotics and automation (ICRA), pp. 9798–9805. IEEE, 2019. B
626 627 628	John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. <i>ArXiv preprint</i> , abs/1707.06347, 2017. URL https://arxiv.org/abs/1707.06347. 2, 4.2
629 630	Karl Sims. Evolving virtual creatures. In <i>Proceedings of the 21st annual conference on Computer graphics and interactive techniques</i> , pp. 15–22, 1994. 1
632 633 634	Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033. IEEE, 2012. 4.1
635 636 637 638 639 640 641	Elise van der Pol, Daniel E. Worrall, Herke van Hoof, Frans A. Oliehoek, and Max Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan- Tien Lin (eds.), Advances in Neural Information Processing Systems 33: Annual Con- ference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/ 2be5f9c2e3620eb73c2972d7552b6cb5-Abstract.html. B
642 643 644	Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural networks. In <i>International conference on learning representations</i> , 2018. B
645 646 647	Tingwu Wang, Yuhao Zhou, Sanja Fidler, and Jimmy Ba. Neural graph evolution: Towards efficient automatic robot design. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL https: //openreview.net/forum?id=BkgWHnR5tm. B

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5028-5037, 2017. B Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris M. Kitani. Transform2act: Learning a transform-and-control policy for efficient agent design. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022. URL https://openreview.net/forum?id=UcDUxjPYWSr. 1, 2, 3.1, 3.1, 3.2, 4.1, 4.2, B, E.3, E.4

702 A PROOFS

In this section, we theoretically prove that our proposed EquiEvo ensures the co-evolution process
 preserve the symmetry as desired.

Finally, we transform the subequivariant graph of the agent's state into an invariant graph

Theorem 1. The LRF canonicalization, denoted as $\mathcal{G}_{\vec{g}} \leftarrow \text{LRF}(\vec{\mathcal{G}}_{\vec{g}})$ is $O_g(3)$ -invariant, satisfying any transformation $g \in O_{\vec{g}}(3)$, $\mathcal{G}_{\vec{g}} = \text{LRF}(g \cdot \vec{\mathcal{G}}_{\vec{g}})$.

710

720 721 722

729

730 731 732

747 748 749

750

752 753

755

711 *Proof.* Let the projected graph $\mathcal{G}_{\vec{g}}$ be the output of the LRF canonicalization with input $\vec{\mathcal{G}}_{\vec{g}} = (\mathcal{G}_{\mathrm{m}}, \mathbf{H}^{\mathrm{b}}, \vec{Z}, \vec{g})$. Similarly, let $\mathcal{G}'_{\vec{g}}$ be the output of the LRF canonicalization with input $g \cdot \vec{\mathcal{G}}_{\vec{g}} = (\mathcal{G}_{\mathrm{m}}, \mathbf{H}^{\mathrm{b}}, g \cdot \vec{Z}, \vec{g})$, for any transformation $g \in \mathrm{O}_{\vec{g}}(3)$. If $\mathcal{G}_{\vec{g}} = \mathcal{G}'_{\vec{g}}$, this indicates that the LRF canonicalization preserves $\mathrm{O}_{\vec{g}}(3)$ -invariance.

We first prove that OP in Equation (6) is equivariant.

Let g be a transformation in $O_{\vec{g}}(3)$, which includes rotation and reflection R along the direction of \vec{g} . Specifically, the transformation is applied as follows:

$$\boldsymbol{O}' \leftarrow g \cdot \boldsymbol{O} \coloneqq R\boldsymbol{O}_i,\tag{15}$$

$$\boldsymbol{z}' \leftarrow \boldsymbol{g} \cdot \boldsymbol{v} \coloneqq R \boldsymbol{z}.$$
 (16)

723 We use the fact that $\forall R \in O_{\vec{g}}(3), \forall \vec{x}, \vec{y} \in \mathbb{R}^3, \langle R\vec{x}, R\vec{y} \rangle = \vec{x}^\top R^\top R \vec{y} = \vec{x}^\top \vec{y} = \langle \vec{x}, \vec{y} \rangle$ and 724 $||R\vec{x}|| = \langle R\vec{x}, R\vec{x} \rangle = \langle \vec{x}, \vec{x} \rangle = ||\vec{x}||$. By definition, $R\vec{e}_3 = \vec{e}_3$. According to the property of 725 Equation (5), we know that $\vec{z}_1^{(L)}, \vec{u}$ and \vec{v} are $O_{\vec{g}}(3)$ -equivariant.

⁷²⁶ ⁷²⁷ Let $O' \leftarrow OP(g \cdot \vec{u}, g \cdot \vec{v})$. By the properties of the orthogonalization process in Equations (9) to (11), we have:

$$\vec{e}'_{3} = [0, 0, 1]^{\top} = \vec{e}_{3} = R\vec{e}_{3}, \tag{17}$$

$$\vec{e}_{1}' = \frac{Ru - \langle R\vec{u}, \vec{e}_{3} \rangle \vec{e}_{3}}{\|R\vec{u} - \langle R\vec{u}, \vec{e}_{3}' \rangle \vec{e}_{3}'\|}$$
(18)

$$= \frac{R\boldsymbol{u} - \langle R\boldsymbol{u}, R\boldsymbol{e}_3 \rangle R\boldsymbol{e}_3}{\|R\boldsymbol{u} - \langle R\boldsymbol{u}, R\boldsymbol{e}_3 \rangle R\boldsymbol{e}_3\|}$$
(19)

$$=\frac{R(\boldsymbol{u}-\langle \boldsymbol{u},\boldsymbol{e}_{3}\rangle\boldsymbol{e}_{3})}{\|R(\boldsymbol{u}-\langle \boldsymbol{u},\boldsymbol{e}_{3}\rangle\boldsymbol{e}_{3})\|}$$
(20)

$$= R \frac{(\vec{u} - \langle \vec{u}, \vec{e}_3 \rangle \vec{e}_3)}{\|(\vec{u} - \langle \vec{u}, \vec{e}_3 \rangle \vec{e}_3)\|}$$
(21)

$$= R\vec{e}_1, \tag{22}$$

$$\vec{e}_{2}' = \frac{R\vec{v} - \langle R\vec{v}, \vec{e}_{1}' \rangle \vec{e}_{1}' - \langle R\vec{v}, \vec{e}_{3}' \rangle \vec{e}_{3}'}{\|R\vec{v} - \langle R\vec{v}, \vec{e}_{1}' \rangle \vec{e}_{1}' - \langle R\vec{v}, \vec{e}_{3}' \rangle \vec{e}_{3}'\|}$$
(23)

$$=\frac{R\vec{v}-\langle R\vec{v},R\vec{e}_1\rangle R\vec{e}_1-\langle R\vec{v},R\vec{e}_3\rangle R\vec{e}_3}{\|R\vec{v}-\langle R\vec{v},R\vec{e}_1\rangle R\vec{e}_1-\langle R\vec{v},R\vec{e}_3\rangle R\vec{e}_3\|}$$
(24)

$$\begin{array}{ccc} 744 & \|R\vec{v} - \langle R\vec{v}, R\vec{e}_1 \rangle R\vec{e}_1 - \langle R\vec{v}, R\vec{e}_3 \rangle R\vec{e}_3 \| \\ 745 \\ 746 & = \frac{R(\vec{v} - \langle \vec{v}, \vec{e}_1 \rangle \vec{e}_1 - \langle \vec{v}, \vec{e}_3 \rangle \vec{e}_3)}{\|R(\vec{v} - \langle \vec{v}, \vec{e}_1 \rangle \vec{e}_1 - \langle \vec{v}, \vec{e}_3 \rangle \vec{e}_3)\|} \end{array}$$

$$(25)$$

$$\|R(\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{e}_1 \rangle \boldsymbol{e}_1 - \langle \boldsymbol{v}, \boldsymbol{e}_3 \rangle \boldsymbol{e}_3)\| \\ - \frac{1}{R} \frac{(\vec{\boldsymbol{v}} - \langle \vec{\boldsymbol{v}}, \vec{\boldsymbol{e}}_1 \rangle \vec{\boldsymbol{e}}_1 - \langle \vec{\boldsymbol{v}}, \vec{\boldsymbol{e}}_3 \rangle \vec{\boldsymbol{e}}_3)}{(26)}$$

$$= R \frac{\vec{v}_{1}}{\|(\vec{v} - \langle \vec{v}, \vec{e}_{1} \rangle \vec{e}_{1} - \langle \vec{v}, \vec{e}_{3} \rangle \vec{e}_{3})\|}$$

$$= R \vec{e}_{2}.$$
(20)
(21)

$$= R \boldsymbol{e}_2. \tag{2}$$

751 Therefore,

$$\mathbf{O}' = [\vec{e}'_1, \vec{e}'_2, \vec{e}'_3] = R[\vec{e}_1, \vec{e}_2, \vec{e}_3] = R\mathbf{O}.$$
(28)

754 Then we have

$$\boldsymbol{z}_{i}' = \boldsymbol{O}'^{\top} \boldsymbol{\bar{z}}_{i}^{(0)'} = \boldsymbol{O}^{\top} \boldsymbol{R}^{\top} \boldsymbol{R} \boldsymbol{\bar{z}}_{i}^{(0)} = \boldsymbol{O}^{\top} \boldsymbol{\bar{z}}_{i}^{(0)} = \boldsymbol{z}_{i}.$$
(29)

 $\mathcal{G}'_{\vec{a}} = (\mathcal{G}_{\mathrm{m}}, \boldsymbol{H}^{\mathrm{b}}, \boldsymbol{Z}', \vec{g})$ (30)

$$= (\mathcal{G}_{\mathrm{m}}, \boldsymbol{H}^{\mathrm{b}}, \boldsymbol{Z}, \boldsymbol{\vec{g}})$$
(31)

$$=\mathcal{G}_{\vec{q}}.$$
(32)

Corollary 1. Let a_t^m, a_t^b, V_{ϕ} be output of the actor and the critic of EquiEvo with \mathcal{G}^m and $\vec{\mathcal{G}}_{\vec{g}}$ as input. Let $a_t^{m'}, a_t^{b'}, V_{\phi}'$ be the actor and critic with $g \cdot \mathcal{G}^m$ and $g \cdot \vec{\mathcal{G}}_{\vec{g}}$ as input, $g \in O_{\vec{g}}(3)$. Then, $(a_t^{m'}, a_t^{b'}, V_{\phi}') = (a_t^m, a_t^b, V_{\phi})$, indicating the EquiEvo preserve $O_{\vec{g}}(3)$ -invariance.

Proof. First, based on the geometric transformation properties of the topology graph, we have

$$\mathcal{G}^{\mathrm{m}\prime} = g \cdot \mathcal{G}^{\mathrm{m}} = \mathcal{G}^{\mathrm{m}}.$$
(33)

Then, by Theorem 1, $\forall R \in \mathbf{O}_{\vec{q}}(3)$, we have

$$\mathcal{G}'_{\vec{g}} = g \cdot \vec{\mathcal{G}}_{\vec{g}} = \mathcal{G}_{\vec{g}}.$$
(34)

Therefore,

$$\boldsymbol{a}_{t}^{\mathrm{m}\prime} = \varphi_{\theta}^{\mathrm{m}}(\mathcal{G}^{\mathrm{m}\prime}) \tag{35}$$

$$=\varphi_{\theta}^{\mathrm{m}}(\mathcal{G}^{\mathrm{m}})=\boldsymbol{a}_{t}^{\mathrm{m}},\tag{36}$$

$$\boldsymbol{a}_t^{\mathrm{b}\prime} = \varphi_{\theta}^{\mathrm{b}}(\mathcal{G}_{\vec{\boldsymbol{a}}}) \tag{37}$$

$$=\varphi_{\theta}^{\mathrm{b}}(\mathcal{G}_{\vec{g}}) = \boldsymbol{a}_{t}^{\mathrm{b}},\tag{38}$$

and

$$V_{\phi}' = \varphi_{\phi}^{\rm b}(\mathcal{G}_{\vec{g}}') \tag{39}$$

$$=\varphi_{\phi}^{\mathbf{b}}(\mathcal{G}_{\vec{g}}')=V_{\phi}.$$
(40)

B RELATED WORK

Optimization of Agent morphology Design: Research in the field of optimization of agent mor-phology design mainly falls into two mainstream approaches: the first focuses on optimizing the properties, functions, and design parameters of agents (Schaff et al., 2019; Ha, 2019; Chen et al., 2023b), and the second on structure optimization based on graph neural networks (Wang et al., 2018; Yuan et al., 2022; Hu et al., 2023). The former often adopts population-based methods, handling morphology and control strategy optimization separately through a two-level optimiza-tion process (Ha, 2019), such as the application of evolutionary search in morphology optimiza-tion (Gupta et al., 2021; Wang et al., 2019); the latter focuses on multi-objective joint optimiza-tion (Yuan et al., 2022). Research also includes the optimization of continuous design parameters of agents, using methods such as simulated annealing, trajectory optimization, and deep reinforce-ment learning (Baykal & Alterovitz, 2017; Ha et al., 2017; Chen et al., 2020). Recent studies have further simplified the evolutionary process and improved task-independent design optimization (Ch-eney et al., 2014; Desai et al., 2017; III et al., 2021). Despite the progress made by these methods, sample efficiency remains a challenge, as agents with different design morphologies often learn independently within the population. Yuan et al. (2022) introduce Transform2Act, an innovative policy for simultaneous morphology design and control optimization that adapts skeletal structures and joint attributes for efficient environmental interaction, advancing integrated agent design and control. Huang et al. (2024) presents CompetEvo, a system that integrates morphology evolution with adversarial game self-training to enhance agents' capabilities, showing morphology evolution significantly boosts combat skills and emergent behaviors.

756 Hence,

810 Geometric Equivariant Models: Notably, the physical realm exhibits inherent symmetries, and 811 extensive research has been conducted on group equivariant models (Cohen & Welling, 2016a;b; 812 Worrall et al., 2017). In recent times, the research area of geometrically equivariant graph neu-813 ral networks (Han et al., 2022b) has emerged, utilizing symmetry as an inductive bias during the learning process. These models are crafted to ensure that their outputs undergo rotation, transla-814 tion, or reflection in alignment with the inputs, thereby preserving the symmetry. Within a Markov 815 decision process (MDP) that exhibits symmetries (van der Pol et al., 2020), the state-action space 816 possesses inherent symmetries, allowing policies to be optimized within a more straightforward, 817 abstract MDP. van der Pol et al. (2020) endeavors to train policy networks that are equivariant and 818 value networks that are invariant in two-dimensional simulated environments. In contrast, Chen et al. 819 (2023a) delves into the study of body-level equivariant policy networks within intricate 3D physics 820 environments, enhancing policy generalization over various orientations. Additionally, Chen et al. 821 (2024) address the challenges of multi-agent games by employing entity assignment along with an 822 entity-level subequivariant message-passing mechanism. Both our approach and Chen et al. (2023a; 823 2024) implement equivariance based on the core function described in Equation (2). However, there 824 are key differences in the design and application. Chen et al. (2023a) employs a transformer archi-825 tecture with a fully connected topology, whereas both Chen et al. (2024) and our approach utilize GNNs. The main distinction lies in the graph construction: Chen et al. (2024) models multi-agent re-826 lationships, requiring an entity assignment to define the graph's topology, while our method focuses 827 on single-agent morphology evolution, directly using the morphology's inherent topology to build 828 the graph. This design aligns with our goal of co-optimizing morphology and behavior efficiently 829 within a single-agent framework. 830

C SUBEQUIVARIANT MESSAGE PASSING

Concretely, we have:

$$\vec{z}_{ij}^{(l)} = [(\vec{p}_j - \vec{p}_i), \vec{z}_i^{(l)}, \vec{z}_j^{(l)}],$$
(41)

$$\boldsymbol{h}_{ij}^{(l)} = [\|\vec{\boldsymbol{p}}_j - \vec{\boldsymbol{p}}_i\|_2, \boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}], \tag{42}$$

$$\vec{m}_{ij}^{(l)}, m_{ij}^{(l)} = \phi_{\vec{g}} \left(\vec{z}_{ij}^{(l)}, h_{ij}^{(l)} \right),$$
(43)

$$\vec{m}_{i}^{(l)}, m_{i}^{(l)} = \sum_{j \in \mathcal{N}(i)} \vec{m}_{ij}^{(l)}, \sum_{j \in \mathcal{N}(i)} m_{ij}^{(l)},$$
(44)

$$(\vec{z}_i^{(l+1)}, \boldsymbol{h}_i^{(l+1)}) = (\vec{z}_i^{(l)}, \boldsymbol{h}_i^{(l)}) + \psi_{\vec{g}} \left([\vec{m}_i^{(l)}, \vec{z}_i^{(l)}], [\boldsymbol{m}_i^{(l)}, \boldsymbol{h}_i^{(l)}] \right),$$
(45)

where

- \vec{p}_i , the position of node *i*;
- + $m{h}_i^{(l)}$ and $m{z}_i^{(l)}$, the scalar and vector states of node i at the l-th layer;

$$h_{ij}^{(l)}$$
 and $\vec{z}_{ij}^{(l)}$, the scalar and vector states of edge ij at the *l*-th layer;

- + $ec{m}_{ij}^{(l)}$ and $m_{ij}^{(l)}$, the vector and scalar messages passed through edge ij;
- $\vec{m}_i^{(l)}$ and $m_i^{(l)}$, the aggregated vector and scalar messages at node *i*;
- $\phi_{\vec{g}}$ and $\psi_{\vec{g}}$, as described in Equation (2), the subequivariant functions used for message passing and state updates.
- 853 854 855 856

857

858

859 860 861

831

832 833

834 835

836 837

838 839 840

841 842 843

844 845

846

847 848 849

850 851

852

D THE ROLE OF GEOMETRIC SYMMETRY IN MORPHOLOGY EVOLUTION

The optimization of morphology in co-evolution tasks is governed by a bi-level objective, as defined in Equation (1):

$$\mathcal{G}_{\mathrm{m}}^* \coloneqq rg\max_{\mathcal{G}_{\mathrm{m}}} J(\pi_{\mathcal{G}}, \mathcal{G}_{\mathrm{m}}) \quad \mathrm{s.t.} \quad \pi_{\mathcal{G}} = rg\max_{\pi} J(\pi, \mathcal{G}_{\mathrm{m}}),$$

where $J(\pi, \mathcal{G}_m)$ represents the cumulative reward for a given morphology \mathcal{G}_m and policy π . This two-layer optimization highlights the dependence of morphology optimization on the behavior policy that performs for a given morphology. It is important to note that J is defined as an expectation, meaning that all states are integrated out in the expectation sense. Consequently, the morphology's policy and value functions theoretically depend only on the morphology \mathcal{G}_{m} and the behavior policy $\pi_{\mathcal{G}}$.

In practice, the bi-level optimization is implemented as a co-optimization process, where the inner 868 and outer optimizations alternate incrementally rather than fully optimizing inner layer. Since the evaluation of J relies on finite samples of the behavior policy, if the behavior policy is not equiv-870 ariant, the limited sampling will cause J to vary under sampled initial state. Geometric symmetry 871 reduces the variability introduced by sampling, as equivariant networks ensure equivalent actions 872 and consistent values, providing consistent feedback across state transformations----such as trans-873 lations, rotations, or reflections. For instance, under rotational symmetry, equivalent actions are 874 treated as identical, and the feedback provided to the morphology is unaffected by differences in sampled initial states. This consistent feedback not only improves the reliability of morphology 875 evaluation but also enhances the efficiency of the co-evolution process, allowing for better optimiza-876 tion within the search space. 877



Figure 11: Visualization of Morphological Evolution Across Training Progress.

Figure 11 compares the morphological evolution during training under two scenarios: without (left) and with (right) the integration of geometric symmetry. Checkpoints at 25%, 50%, 75%, and 100% of the training process are visualized for two random seeds (s0 and s1). Without injecting geometric symmetry (left), morphology evolution appears asymmetric and incomplete, with structural transformations seemingly getting trapped in local optima early in the process, leading to lower reward curves. This may be attributed to the nature of reinforcement learning, where morphology and behavior are gradually evolved through abundant exploration of the environment. During this learning process, the co-evolution of morphology and behavior becomes highly vulnerable to local minima, and searching for a good policy within the large space would be notoriously difficult. **In contrast, injecting geometric symmetry (right) fosters more intricate and comprehensive morphology evolution, accompanied by higher reward curves, indirectly supporting the claim that geometric symmetry enhances the efficiency of the co-evolution process by compacting the search space redundancy in a lossless manner, enabling better optimization.**

E MORE EXPERIMENTALS DETAILS AND RESULTS

E.1 REWARD

878 879 880

883

885

887

889

890

891

892

893

894

895

896

897

898

899

900 901 902

903 904

905 906

Humanoid Navigation: For the navigation task, we design the reward for Humanoid agent. The de-907 signed reward structure comprises six components: a. Success Bonus: A significant sparse reward 908 of 1,000 is awarded. **b. Forward Reward:** It is quantified as $1.25 \times \frac{\vec{p}}{|\vec{p}||} \Delta \vec{p}$, where $\frac{\vec{p}}{|\vec{p}||}$ represents 909 910 the unit direction vector of the agent's position, and $\Delta \vec{p}$ represents the displacement of the agent 911 from the current time step. c. Distance Reward: A dense reward to incentivize the achievement of 912 the task objective. It is computed as $\frac{\Delta s}{\Delta t}$, where Δs represents the change in distance from the agent to the target. **d. Control Cost:** This penalty discourages agents from executing excessively large actions, and is calculated as $-0.1 \times \sum \sqrt{a_i}$. **e. Contact Cost:** The penalty discourages excessive 913 914 contact forces, and is calculated as $-\min(0.5 \times 10^{-6} \times \sum c_i^2, 10)$, where c_i represents the compo-915 nent of the force exerted on the agent by external forces. f. Survive Reward: For each step taken, 916 the agent receives a survival reward of 5. Thus, the reward is equal to the sum of the aforementioned 917 rewards.

Ant Navigation: For the navigation task, we design the reward for Ant agent. The designed reward structure comprises four components: **a. Success Bonus:** A significant sparse reward of 1,000 is awarded. **b. Distance Reward:** A dense reward to incentivize the achievement of the task objective. It is computed as $10 \times \frac{\Delta s}{\Delta t}$, where Δs represents the change in distance from the agent to the target. **c. Control Cost:** This penalty discourages agents from executing excessively large actions, and is calculated as $-0.01 \times \sum \sqrt{a_i}$. **d. Survive Reward:** For each step taken, the agent receives a survival reward of 1. Thus, the reward is equal to the sum of the aforementioned rewards.

925 Ants Sumo: For Ants Sumo task, we design the reward structure. The designed reward structure 926 comprises seven components: a. Win Bonus: A sparse reward of 2000 for achieving the win 927 condition. b. Lose Bonus: A sparse penalty of -2000 for the losing condition. c. Draw Bonus: A 928 sparse penalty of -1000 for the draw condition. This penalty is meant to encourage confrontation. d. Control Cost: This penalty discourages agents from executing excessively large actions, and 929 is calculated as $-0.1 \times \sum \sqrt{a_i}$. e. Push Opponent Reward: This motivates agents to push the 930 opponent, calculated as $-10 \times \exp(-||\vec{p}_{opp}||)$, where \vec{p}_{opp} represents the position of the opponent. **f.** 931 **Move Reward:** This motivates agents to move closer to the opponent, calculated as $10 \times \max(\Delta \vec{p} \cdot \Delta \vec{p})$ 932 $\frac{\vec{p}_{opp}-\vec{p}}{||\vec{p}_{opp}-\vec{p}||}, 0)$, where $\Delta \vec{p}$ represents the displacement of the agent from the current time step. g. 933 Survive Reward: For each step taken, the agent receives a survival reward of 2. Thus, the reward 934 is equal to the sum of the aforementioned rewards. 935

936 937

938

E.2 BASELINE METHODS

Ant Navigation: In the navigation task, we use Ant agent for experimentation. The agent we
improve by using EquiEvo method is named EquiEvoAnt, which integrates both subequivariance
and morphology evolution mechanism. To conduct a comprehensive comparison, we define the
following three baseline methods: EquiAnt: This baseline method maintains subequivariance but
removes the morphology evolution component. EvoAnt: This baseline method includes morphology
evolution but does not integrate subequivariance. Ant: This is the most basic baseline method,
lacking both subequivariance and morphology evolution component.

946 **Humanoid Navigation:** In the navigation task, we also introduce Humanoid agent for experimentation, while we omit structure transform stage during evolution. After applying the EquiEvo method, 947 we name the agent EquiEvoHumanoid. To conduct a comprehensive comparison, we define the 948 following three baseline methods: EquiHumanoid: This baseline method maintains subequivariance 949 but removes the morphology evolution component. EvoHumanoid: This baseline method includes 950 morphology evolution but does not integrate subequivariance. Humanoid: This is the most basic 951 baseline method, lacking both subequivariance and morphology evolution capabilities. In this task, 952 we skip structure transform stage. 953

Ants Sumo: In the sumo task, we still use the Ant agent for experimentation. We adopt the EquiEvo method and omit the structure transform stage during evolution, naming it EquiEvoAnt, which integrates both subequivariant networks and morphology evolution mechanisms. To conduct a comprehensive comparison, we define the following three baseline methods: EquiAnt: This baseline method maintains subequivariance but removes the morphology evolution component. EvoAnt: This baseline method includes morphology evolution but does not integrate subequivariance. Ant: This is the most basic baseline method, lacking both subequivariance and morphology evolution capabilities. In this task, we skip structure transform stage.

961 962

963

E.3 ANALYSIS OF MORPHOLOGY-BEHAVIOR MAPPING

964 We conduct experiments on the Ant Navigation task to analyze the impact of Morphology-Behavior 965 Mapping, particularly focusing on the role of JSMLP, which was proposed in Transform2Act (Yuan 966 et al., 2022) to differentiate joint actions. While JSMLP introduces body-specific characteristics, 967 enhancing effectiveness in some contexts, it reduces the generality of the morphology-behavior 968 mapping and impedes reusability in tasks with symmetric structures. To validate this, we ablate four variants: Evo w. JSMLP (baseline with JSMLP), Evo w.o. JSMLP (baseline without JSMLP), 969 EquivEvo w.o. JSMLP (our method without JSMLP), and EquivEvo w. JSMLP (our method with 970 JSMLP). As shown in Figure 12, the removal of the JSMLP module improves performance, with 971 EquivEvo w.o. JSMLP outperforming all variants. This indicates that eliminating JSMLP enhances



Figure 12: Analyses of Morphology-Behavior.

adaptability and learning by maintaining the generality of the morphology-behavior mapping, allowing better exploitation of task symmetry. Consequently, simplifying the model by omitting JSMLP within the *EquivEvo* framework facilitates more efficient navigation in symmetric tasks.

E.4 DISCUSSION ON STRUCTURAL EVOLUTION

Structural evolution is a promising yet challenging direction in morphology-behavior cooptimization. In this subsection, we provide a detailed discussion of our decisions, experimental 995 findings, and future directions regarding structural evolution. 996

Evolving from Scratch and Predefined Morphology Structures. For the Ant Navigation task, 998 we chose to evolve morphology skeletons from scratch, starting from the "atomic morphology" 999 shown in Figure 1. This setup represents the most challenging scenario in co-evolution, allowing us 1000 to demonstrate the effectiveness of our method while avoiding biases introduced by predefined mor-1001 phologies. In contrast, the Ants Sumo and Humanoid Navigation tasks use predefined morphology 1002 structures, where only attribute development is allowed. This decision are driven by the challenge 1003 of defining clear boundaries for structural transformations across different starting morphologies, 1004 particularly for complex predefined morphology structures.

1005

972 973

974 975

976

977 978

979

980

981

982

983

984 985

986 987 988

989

990

991 992

993

994

997

Challenges of Structual Evolution. Structural evolution is inher-1007 ently complex and remains underexplored. Several key issues must be addressed, such as how many body parts can grow from a torso 1008 node, how many can branch from each node, which nodes require 1009 constraints, the maximum structural depth, and the corresponding 1010 attribute transformations. Without carefully designed constraints, 1011 the search space grows exponentially, making optimization ineffi-1012 cient and unstable. 1013

- 1014
- Structual Evolution for Ants Sumo. In the EvoAnts Sumo task 1015 (see Figure 13), structural evolution can lead to uncoordinated de-



Figure 13: EvoAnts Sumo.

1016 velopment of multiple body parts, resulting in excessive forces that cause the agent to fling itself 1017 out of the arena. This instability prevents meaningful results, which is why we only included the 1018 outcomes of attribute evolution for this task.

1019

1020 Structural Evolution for Humanoids. Training humanoids is significantly more challenging 1021 compared to simpler morphologies like the ant. Achieving stable behavior in attribute evolution already requires extensive training and computational resources. Adding structural evolution to the mix would drastically increase complexity, demanding even longer training times and greater 1023 computational resources. Given our current computational limitations, we are unable to conduct ex-1024 periments involving structural evolution for humanoids. Nonetheless, we acknowledge the potential 1025 value of this research direction and will pursue it in future work.

1026 The Complexity of 3D Structural Evolution. Incorporating structural evolution into 3D tasks is 1027 exceptionally challenging. The interplay between the dimensionality of the environment, task, and 1028 action space-including morphology structure, attributes, and behaviors-creates a compounded 1029 effect that leads to an exponential explosion in the optimization search space. Furthermore, such a 1030 problem setup has not been systematically explored before. Prior work has typically focused on 2D spaces (Ha, 2019), 3D spaces with 2D structural tasks (Yuan et al., 2022; Gupta et al., 2021; Dong 1031 et al., 2023), tasks that do not incorporate structural evolution (Huang et al., 2024), or have imposed 1032 rigid constraints on morphology, such as bilateral symmetry (Dong et al., 2023), limiting the natural 1033 development of morphology through interaction with the environment. 1034

Our work represents a crucial step forward by leveraging geometric symmetries to reduce this complexity. We believe our method lays the foundation for exploring this valuable and fascinating problem setup. While our current results are a small step, they highlight the necessity and significance of addressing these challenges to advance the field further.

1039

1046

Future Directions. The limitations and insights gained from our experiments point to several exciting future directions. These include extending structural evolution to more complex morphologies like humanoids, refining structural constraints to balance exploration and optimization, and systematically investigating the role of geometric symmetries in 3D structural tasks. By continuing to explore these directions, we aim to address the challenges of structural evolution and unlock its full potential for embodied intelligence and robotic design.

1047	Tuble 2. Hyperparameter bettings for E	quittio
1048	Hyperparameter	Setting
1049	Number of structural transformations N_c	5
1050	Number of attribute transformations N_{2}	1
1051	Number of control executions N_e	1.000
1052	Topological graph neural network layer type	GraphConv
1053	Geometric graph neural network layer type	SubequivariantMl
1054	Topological graph neural network size	
1054	(structural transformation phase)	(64, 64, 64)
1055	Topological graph neural network size	
1056	(attribute transformation phase)	(64, 64, 64)
1057	Geometric graph neural network vector size	
1058	(geometric subgroup equivariant transformation phase)	(16, 16, 16)
1059	Geometric graph neural network scalar feature size	
1060	(geometric subgroup equivariant transformation phase)	(32, 32, 32)
1000	Topological graph neural network size	
1001	(control execution phase)	(64, 64, 64)
1062	Policy learning rate	5e-5
1063	Value estimation graph neural network size	(64, 64, 64)
1064	Value estimation MLP size	(512, 256)
1065	Value learning rate	3e-4
1066	PPO clipping ϵ	0.2
1000	PPO batch size	50000
1007	PPO mini-balch size	2048
1068	Troining apochs	1000
1069	Discount factor of	0.005
1070	$GAE \lambda$	0.995
1071		0.95
1072		
1073		

 Table 2: Hyperparameter Settings for EquiEvo

1076

1077

1078

Ala	onithm 1 EquiEvo
	Initialize policy π_{a} and value V_{a} (including topological graph neural networks (a^{s} (a^{a} (a^{b}))
· 1. 2	and subequivariant graph neural network φ_{θ}^{g} {including topological graph neural networks $\varphi_{\theta}, \varphi_{\theta}, \varphi_{\theta}, \varphi_{\theta}$, and subequivariant graph neural network φ_{θ}^{g} }
3 2:	while not reached maximum iteration do
3:	Initialize memory $\mathbb{M} \leftarrow \emptyset$
; 4:	while \mathbb{M} not reached batch size do
5:	$\mathcal{G}^{\mathrm{m}} \leftarrow \mathrm{initial} \ \mathrm{morphology} \ \mathrm{topology} \ \mathrm{graph}$
7	{Structure Transform Stage}
6:	for $t = 0, 1,, N_s - 1$ do
7:	Sample structure transform action $a_t^s \sim \varphi_{\theta}^s(\mathcal{G}^m)$
8:	$\Phi \leftarrow \text{Structure}$
9: 1 10	$\mathcal{G}^{\mathrm{m}} \leftarrow \text{apply } a^{\mathrm{s}} \text{ to modify } \mathcal{G}^{\mathrm{m}} \text{ s morphology structure } (\mathcal{V}, \mathcal{E})$
	$r \leftarrow 0$; Store $(r, a^{\circ}, \mathcal{G}^{\circ}, \Phi)$ to IVI
- 11:	ena ior
12.	{Autobuce transform stage} for $t = N$ $N \pm N = 1$ do
= 12. □ 13·	Sample attribute transform action $a^a \sim c^a(\mathcal{C}^m)$
b 13. b 14.	$\Phi \leftarrow \text{Attribute}$
- 15:	$\mathcal{G}^{\mathrm{m}} \leftarrow \text{apply } a^{\mathrm{a}}$ to modify \mathcal{G}^{m} 's bone joint attributes H^{m}
16:	$r \leftarrow 0$: Store $(r, a^{a}, \mathcal{G}^{m}, \Phi)$ to M
17:	end for
1	{Behavior Control Stage}
18:	$\vec{\mathcal{G}}_{\vec{n},N+N} \leftarrow \text{initial morphology subequivariant graph}$
19:	for $t = N_{\rm s} + N_{\rm a}, \dots, N_{\rm s} + N_{\rm a} + N_{\rm e} - 1$ do
2 20:	$\vec{v}_t \leftarrow \varphi^{\mathrm{g}}_{\varrho}(\vec{\mathcal{G}}_{\vec{q}}) \{ \text{LRF Canonicalization} \}$
³ 21:	Use the calculated LRF transformation vector \vec{u}, \vec{v} to predict a local reference frame a
4	project the morphology subequivariant graph $\vec{G}_{\vec{n}}$ to the geometry-invariant represent
5	tion $\mathcal{G}_{\vec{a}}$
⁶ 22:	Sample motor control action $a^{\rm b} \sim \varphi_a^{\rm b}(\mathcal{G}_{\vec{a}})$
23:	$\Phi \leftarrow \text{Execution}$
^B 24:	$\mathcal{G}_{\vec{a}} \leftarrow \text{environment dynamics } P^{\mathrm{b}}(\mathcal{G}_{\vec{a}} \mathcal{G}_{\vec{a}}, a^{\mathrm{b}})$
9 25:	$r \leftarrow \text{environment reward; Store } (r, \mathbf{a}^{\mathbf{b}}, \mathcal{G}_{\vec{a}}, \Phi) \text{ to } \mathbb{M}$
⁰ 26:	end for
1 27:	end while{Policy Network Update}
2 28:	Update policy π_{θ} and value V_{ϕ} using PPO algorithm based on samples in M
	end while
3 29:	chu white