

000 MORPHOGEN: EVOLVING ROBOT MORPHOLOGIES 001 WITH LARGE LANGUAGE MODELS 002 003 004

005 **Anonymous authors**

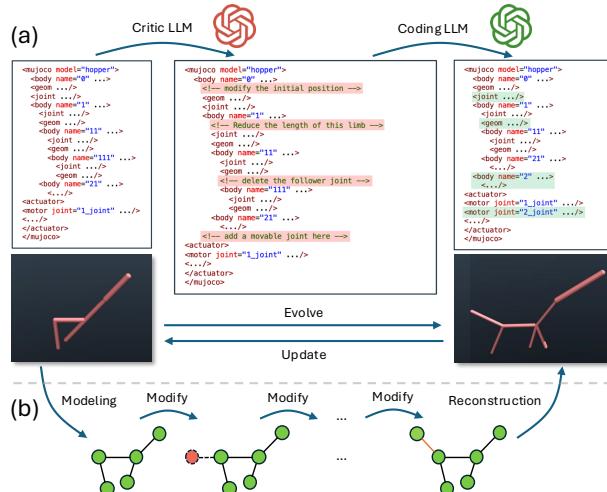
006 Paper under double-blind review

009 ABSTRACT

011 Designing high-performing robot morphologies is a grand challenge for developing
012 specialized autonomous agents. However, the vast, combinatorial, and
013 non-differentiable nature of the morphological design space has been a primary
014 obstacle. Existing methods tackle this problem *indirectly*, relying on either
015 semantically-blind genetic operators or reinforcement learning with predefined
016 modification actions, both of which constrain exploration. In this work, we intro-
017 duce **MorphoGen**, a novel framework that reframes morphological design as a
018 code generation problem. MorphoGen leverages large language models (LLMs)
019 to *directly* iterate the XML files as codes that define an agent’s morphology, solv-
020 ing the original open problem without being limited by any prior constraints or
021 fixed action spaces. Gradient-like textual guidance is provided to steer the evo-
022 lution of robot morphologies through prompted mutations and crossovers. Our
023 approach allows the LLMs to apply its understanding of structure and syntax to
024 generate complex and semantically coherent design variations, enabling an uncon-
025 strained and efficient exploration of the design space. On a suite of challenging
026 locomotion benchmarks, MorphoGen discovers novel and high-performing mor-
027 phologies, significantly outperforming strong baselines by over 52.9% in down-
028 stream motoring evaluation. Our work unlocks a new paradigm for automated
029 robotic design, demonstrating the effectiveness of LLMs in navigating complex,
030 structured engineering search spaces. Codes for our work are released anonym-
031 ously at <https://anonymous.4open.science/r/MorphoGen-ACC/>.

032 1 INTRODUCTION

033 The functionality of a robotic agent
034 is fundamentally determined by its
035 physical form, or morphology (Yuan
036 et al., 2022; Matthews et al., 2023).
037 This morphology is often defined by
038 a structured file, such as the XML
039 format used by MuJoCo (Todorov
040 et al., 2012), which serves as
041 the robot’s genotype specifying the
042 precise configuration of skeletons,
043 joints, and their corresponding at-
044 tributes. Automating the search for
045 an optimal genotype is a highly val-
046 uable endeavor, as manually design-
047 ing and adapting robot morpholo-
048 gies for different tasks is a laborious,
049 intuition-driven process. How-
050 ever, this automation is a non-trivial
051 challenge. The primary difficulty lies
052 in the immense, combinatorial design
053 space; with a nearly infinite number
054 of possible XML definitions, it is im-



055 Figure 1: Comparison between paradigms of robot morphology design. (a) The proposed MorphoGen framework,
056 in which LLMs directly operate on the raw XML represen-
057 tation to generate complete robot morphologies in a single
058 step. (b) The conventional approach, which relies on simpli-
059 fied intermediate models and requires multiple handcrafted
060 modification steps.

possible to exhaustively iterate through all potential skeletal structures and attributes. This challenge is compounded by the high computational cost of evaluation, as each candidate morphology necessitates a costly “inner loop” of control policy optimization to assess its fitness.

To manage this complexity, existing methods avoid operating directly on raw XML definitions, instead introducing an intermediate graph-based abstraction as a mediator. In this common paradigm, a robot’s morphology is represented as a directed graph where topological elements (nodes and edges) denote body parts, joints, and connections with physical attributes (Sims, 2023). Traditional evolutionary algorithms have long navigated this space by applying simplistic, hand-crafted genetic operators, such as adding or deleting nodes and edges (Wang et al., 2019; Sims, 2023; Cheney et al., 2014). However, these operations are semantically blind; they modify the graph’s structure without any inherent understanding of the physical consequences, leading to inefficient exploration. More recent learning-based approaches also build upon this graph abstraction, using graph neural networks (GNN) and reinforcement learning (RL) to learn a policy over a set of predefined graph modification actions (Yuan et al., 2022; Dong et al., 2023). While this learns a more intelligent strategy for applying edits, the edits themselves are still constrained to the same limited set of graph operations. Attempts utilizing LLMs for robot design often fall into similar traps. Some rely on predefined grammar rules that restrict the search to specific families (Qiu et al., 2024), while others focus primarily on parameter tuning within fixed skeletal graphs rather than open-ended topological generation (Fang et al., 2025). In essence, current paradigms sidestep the original, more challenging problem of direct XML generation. By operating within a constrained action space, they simplify the problem at the cost of sacrificing the vast expressiveness and higher degrees of freedom inherent to the true design space of XML definitions, thereby limiting their creative potential.

Recent advancements in large language models (LLMs) have demonstrated their remarkable creative capabilities of generating raw forms of structured language (Achiam et al., 2023; Guo et al., 2025), such as computer codes (Chen et al., 2021). In particular, powerful LLMs combined with automated evaluators have proven effective at evolving and optimizing complex algorithms, as highlighted by the groundbreaking AlphaEvolve project (Novikov et al., 2025; Sharma, 2025). This success unlocks a new possibility for robotic design: directly operating on XML definitions in their original, open form. Inspired by this paradigm, we introduce MorphoGen, a novel framework that treats a robot’s XML genotype as a form of specialized code and reframes morphology design as a code generation problem. Our core idea is to replace the predefined, hand-crafted genetic operators of traditional evolutionary algorithms with an LLM that is prompted to perform semantically rich mutation and crossover operations (Figure 1). To guide this process, we develop a critic LLM that provides gradient-like textual guidance to the primary coding LLM, steering the evolution of robot morphologies. Our approach allows the LLMs to apply their understanding of structure and syntax to directly iterate on raw XML files, enabling an unconstrained and efficient exploration of the design space that lies outside the valid solution spaces of grammar-based methods. To improve the efficiency of evolution, we further propose structure pretraining to serve as high-performing “parent” robots and a two-stage fast controller optimization to accelerate the evaluation of each design. MorphoGen thereby integrates the generative power of LLMs with evolutionary search to design robots in their raw format, unconstrained by the abstractions or limited action spaces of previous methods.

The main contributions of this paper are as follows: (1) We propose an LLM-driven code evolution paradigm for robot design that directly iterates on raw XML definitions of robot morphology. This removes the need for intermediate mediators, thereby addressing the design problem in its original, unconstrained form. (2) Our MorphoGen framework enables efficient morphology optimization through the use of a critic LLM offering gradient-like textual prompts to guide the coding LLM and an accelerated evaluation process for candidate designs. (3) We conduct extensive experiments on challenging locomotion benchmarks, which validate that our approach generates high-performing morphologies that significantly outperform strong baselines by over 52.9% in downstream motoring performance and discovers a diverse range of novel designs.

2 RELATED WORKS

2.1 ROBOT MORPHOLOGICAL DESIGN

Robot morphological design is a complex bilevel optimization problem (Kim et al., 2021; Hu et al., 2023; He & Ciocarlie, 2024; Jiang et al., 2021; Lu et al., 2025; Cheney et al., 2016; Lu et al., 2025;

Pathak et al., 2019), with the outer level optimizing the structure and the inner level refining the control policy for motion. Traditional approaches such as genetic algorithms (Datta et al., 2015; Brodbeck et al., 2015; Liu et al., 2022) and evolutionary strategies (Xu et al., 2021; Cheney et al., 2018; Zhao et al., 2020), rely on heuristic or search-based methods, which are constrained by predefined search spaces and require explicit design rules. Recent RL-based methods (Yuan et al., 2022; Dong et al., 2023; Singh et al., 2022) have advanced control policy optimization for various skeletons but handle structures indirectly through parameterized representations, often imposing constraints for feasibility. In contrast, our approach directly evolves the robot’s skeleton by modifying its underlying XML file, enabling exploration of a broader, unconstrained design space and facilitating the discovery of novel, efficient configurations.

2.2 LLM-BASED EVOLUTIONARY FRAMEWORKS

Recent studies (Tian et al., 2025; Liu et al., 2025; Yang et al., 2023) investigate using LLMs as intelligent search operators in evolutionary optimization, leveraging their capacity for code generation and in-context reasoning. For example, RoboMorph (Qiu et al., 2024) represents robot designs as grammar strings and uses a best-shot prompting approach to iteratively propose new rules. RoboMoRe (Fang et al., 2025) focuses on the co-design of morphology and reward functions and utilizes masked templates for parameter optimization. LASER (Song et al., 2025), incorporate reflection mechanisms to encourage structural diversity. However, these methods fundamentally constrain the search space to a narrow subspace by relying on predefined data structures, and they lack dedicated efficiency optimizations for text-based evolution process. This results in generated morphologies that often exhibit high homogeneity and struggle to surpass the performance of traditional RL-based baselines. To overcomes these challenges, our approach reframes morphology design as unconstrained XML code generation, with a novel critic LLM to provide directional guidance and hierarchical proxy fitness to enhance search efficiency in the complex design space.

3 PRELIMINARIES

3.1 ROBOT MORPHOLOGY

A robot morphology M is formally defined as a Kinematic Tree (Gupta et al., 2022; Liu et al., 2024) $\mathcal{T} = (\mathcal{V}, \mathcal{E}, r)$, where $\mathcal{V} = \{b_0, b_1, \dots, b_n\}$ denotes a set of rigid bodies. Each body b_i is associated with a set of physical attributes Φ_i . The element $r \in \mathcal{V}$ is the root body, representing the base of the kinematic hierarchy. \mathcal{E} denotes a set of directed edges where $e_{ij} = (b_i, b_j)$ represents a kinematic connection from the parent body b_i to the child body b_j . Each edge e_{ij} is associated with a set of parameters $\Theta_{ij} = (J_{ij}, A_{ij})$, where J_{ij} defines the joint properties and A_{ij} defines the actuation properties that control the joint movement. This Kinematic Tree \mathcal{T} can be translated to text-based XML definition \mathcal{X} using a single-valued serialization function \mathcal{F} , such that $\mathcal{X} = \mathcal{F}(\mathcal{T})$.

3.2 EVOLUTIONARY OPTIMIZATION

Begein with an initial morphology \mathcal{X}_0 , the framework proposes candidate modifications to produce successive XML structures \mathcal{X}_t for $t = 1, 2, \dots, T$, where T is the maximum number of iterations. To evaluate the quality of a robot structure \mathcal{X} , a motion controller $\pi_{\mathcal{X}}$ is derived specifically for its morphology. The fitness of \mathcal{X} with $\pi_{\mathcal{X}}$ is assessed in a physics-based simulator, yielding a scalar score $\mathcal{L}(\mathcal{X}, \pi_{\mathcal{X}})$, which quantifies the locomotion efficiency. The robot morphology optimization problem is thus formulated as:

Input: An initial XML structure \mathcal{X}_0 , and an initial motion controller π_0 .

Output: An optimized XML structure \mathcal{X}^* , along with its associated controller $\pi_{\mathcal{X}^*}$.

Objective: Find the morphology $\mathcal{X}^* = \arg \max_{\mathcal{X} \in \Omega_{\mathcal{X}}} \mathcal{L}(\mathcal{X}, \pi_{\mathcal{X}})$ that maximize the fitness.

4 METHOD

We propose MorphoGen, an LLM-based evolutionary framework that directly iterates on the robot’s raw XML genotype to generate high-performing skeletal structures. As illustrated in Figure 2, Mor-

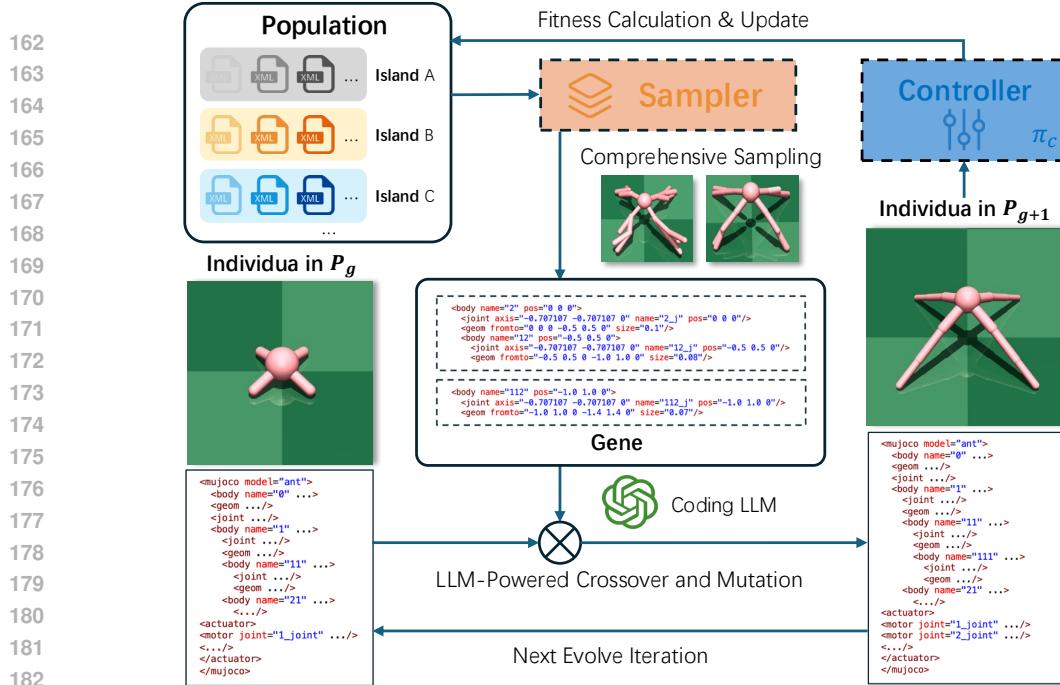


Figure 2: Overview of the MorphoGen evolutionary pipeline.

MorphoGen employs a guided evolutionary search strategy. Begin with initial robot morphologies, a critic LLM is employed to generating textual feedback that critiques their structural strengths and weaknesses. This feedback serves as a rich, gradient-like textual prompt guiding the coding LLM to perform semantically-aware mutation and crossover operations directly on the parent XML genotypes to produce a new generation of optimized offspring morphologies. Subsequently, each newly generated XML genotype is evaluated with an efficient control policy fine-tuning process to determine its fitness. This closed-loop process, driven by textual critique and semantic code generation, transforms the design search from blind and limited modifications into an intelligent, targeted exploration. Consequently, MorphoGen navigates the vast XML definition space in its original, open form with remarkable efficiency, discovering novel, high-performing robot morphologies that surpass the capabilities of traditional methods. Appendix C provides the prompts utilized within MorphoGen.

4.1 LLM DRIVEN EVOLUTIONARY FRAMEWORK

Drawing inspiration from AlphaEvolve (Novikov et al., 2025), the core components of evolution are tailored towards robotic design with redefined XML context as follows:

- **Individua** M is a complete XML definition, which serves as the robot’s genotype by explicitly defining its entire skeletal structure and joints, and physical properties. This genotype is the direct optimization target of our framework.
- **Genes** $\{G_1, G_2, \dots, G_k\} \in M$ are conceptualized as specific and semantically meaningful blocks within the XML definition, which are the core elements manipulated during evolution.
- **Island** $I = \{M_1, M_2, \dots, M_n\}$ is a collection of n such individuals.
- **Population** $P = \{I_1, I_2, \dots, I_m\}$ is a collection of m islands, which contains all the individuals.

Designing robot morphology differs drastically from code generation of AlphaEvolve (Novikov et al., 2025) in every stage of the pipeline, and we propose parent selection, crossover and mutation, as well as fitness calculation, that prioritize morphology understanding, XML genotype modification, and locomotion optimization. In specific, one generation P_g of robot morphology XML definitions are evolved to the next P_{g+1} as follows:

Comprehensive Sampling. To form the parent pool for the subsequent generation from the current population P_g , we employ a hybrid selection strategy designed to balance the exploitation and exploration through a combination of three criteria: (1) **Elitism** (Qiu et al., 2024): The top individuals

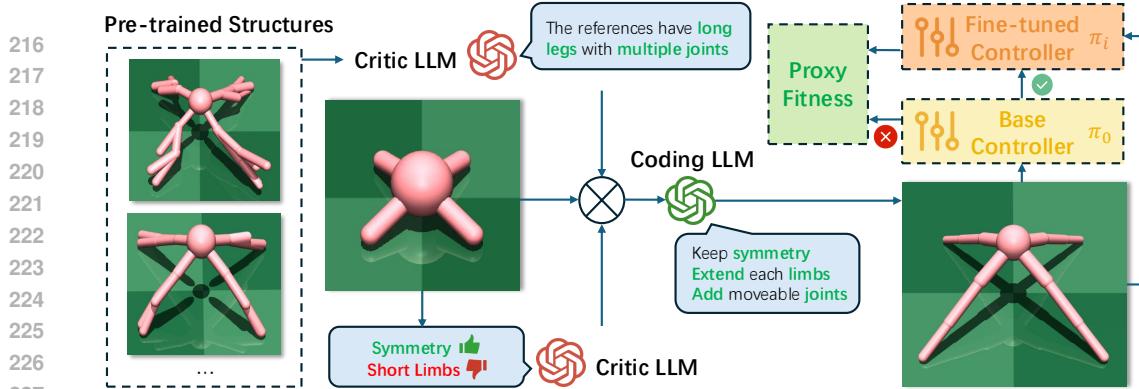


Figure 3: Illustration of the efficient evolutionary optimization process in MorphoGen.

with the highest fitness scores of each island are selected to preserve and propagate high-performing genotypes. (2) **Diversity** (Schulman et al., 2017; Mouret & Clune, 2015): To avoid premature convergence to local optima and promote exploration of diverse regions in the design space, individuals structurally dissimilar to others that maximize Tree Edit Distance $TED(M_i, M_j)$ on the XML structure are chosen. (3) **Randomness** (Fang et al., 2025): A small subset of individuals is chosen uniformly at random to maintain a reservoir of potentially valuable XML blocks not present in high-fitness or diverse candidates. This multi-pronged selection strategy ensures that the evolutionary search is simultaneously greedy, exploratory, and robust.

LLM-Powered Crossover and Mutation. Unlike traditional evolutionary algorithms that rely on simplistic, random operators, MorphoGen leverages a coding LLM to perform semantically coherent XML modifications. In the crossover operation, an offspring M_{child} is generated by intelligently combining structural features from several parent individuals, $\{M_{p_1}, M_{p_2}, \dots, M_{p_i}\}$. The LLM is prompted to merge high-quality XML blocks from all parents, producing a structurally sound offspring that inherits meaningful traits. In the mutation operation, novel variations are introduced into a single parent M_p , modifying its XML genotype to reach better motion performance.

Fitness Calculation. After generating a new XML genotype M , we fix its structure and employ RL algorithm (Yuan et al., 2022) to develop an adaptive controller π_C for locomotion. The RL training process optimizes π_C over a fixed number of steps in a simulated environment, after which the control policy is deployed to evaluate the robot’s performance. The fitness F of the skeletal structure M is quantified by measuring the Euclidean distance traveled from the starting point within a fixed time horizon in the simulation. This evaluation provides a robust indicator of the structure’s effectiveness, leading to guided population update for the next generation. Details of robot controller optimization is provided in Appendix B.

4.2 EFFICIENT EVOLUTIONARY OPTIMIZATION

Evolutionary algorithms are notoriously sample-inefficient, which is even worse in robot design due to the costly inner-loop optimization required to evaluate each skeletal structures. MorphoGen addresses this bottleneck with three key innovations: structure pretraining to initialize the population with high-quality genotypes, text-based gradient guidance to enable targeted optimization, and hierarchical proxy fitness to accelerate evaluation. Together, these components transform the evolutionary search into a more efficient and directed process.

Structure Pretraining. Traditional robot morphology design heavily relies on expert knowledge, and initializing an LLM to explore the expansive structure space from scratch is computationally prohibitive. To overcome this, MorphoGen leverages domain expertise by initializing the population P_0 with a diverse set of high-quality XML genotypes derived from expert-designed morphologies or existing advanced methods. These pre-trained skeletal structures encode rich domain knowledge about effective robot postures and dynamics. Benefiting from this informed gene pool, the coding LLM focuses its exploration on high-potential regions of the design space, significantly enhancing the quality of generated skeletal structures and the efficiency of the evolutionary search.

Text-based Gradient Guidance. Unlike traditional evolutionary algorithms, where random mutations often lead to degraded performance, MorphoGen employs a critic LLM to provide targeted

optimization guidance. For each XML definition, the critic LLM analyzes the skeletal structure and generates textual feedback that identifies the likely structural causes for the motion performance, highlighting both structural strengths and weaknesses. This guidance is then integrated into the prompt for the coding LLM, guiding it to perform semantically-aware mutations that preserve advantageous features while addressing identified deficiencies. This guided approach ensures that each evolutionary step is more likely to be a meaningful improvement, dramatically increasing the directional efficiency of the search.

Proxy Fitness. Learning an optimal control policy for each XML genotype M_i from scratch is computationally expensive, often requiring millions of simulation interactions. However, recognizing that an offspring is often a small modification of its parent, we posit that their optimal control policies are also close in the parameter space, which enables efficient policy reuse. Therefore, instead of full retraining, we can approximate the best control policy of current morphology by briefly fine-tuning the parent’s optimized policy. Moreover, with potentially thousands of generated skeletal structures throughout the whole process, even fine-tuning can be time-consuming. To overcome the above challenges, MorphoGen first employs a basic controller π_0 which is pre-trained for initial robot morphology, and then utilizes a two-stage hierarchical proxy fitness evaluation. Specifically, in the first stage, each skeletal structures M_i are directly controlled with π_0 and we can simply consider the corresponding locomotion scores as the fitness F_i^1 . This evaluation is extremely fast and does not require any additional training costs. We take F_i^1 as an initial screening criterion to ignore those structures that have obvious functional issues. For the skeletal structures with F_i^1 greater than a certain threshold, which indicates a promising candidate, the second stage is triggered. In the second stage, π_0 will be fine-tuned against structure M_i for several steps to get an adaptive controller π_i which is applied to calculate the locomotion score as fitness F_i^2 . The final proxy fitness F_i is thus the result of this conditional, two-stage process. This hierarchical approach drastically reduces the average evaluation time per individual, allowing MorphoGen to assess a much larger volume of candidates and conduct a more thorough and effective exploration of the design space.

In summary, through an integrated framework of LLM-driven evolution, gradient-like textual guidance, and accelerated fitness evaluation, we achieve direct optimization in the raw XML definition space with improved efficiency. We now validate the effectiveness of this approach experimentally.

5 EXPERIMENTS

5.1 EXPERIMENT SETTINGS

Baselines We compare MorphoGen with the following baselines: Neural Graph Evolution (NGE) (Wang et al., 2019), Evolutionary Structure Search (ESS) (Cheney et al., 2018), Random Graph Search (RGS) (Wang et al., 2019), Transform2Act (T2A) (Yuan et al., 2022), Symmetry-Aware Robot Design (SARD) (Dong et al., 2023), RoboMorph (Qiu et al., 2024) and Robo-MoRe (Fang et al., 2025). For the LLM-based methods, we adopt Qwen3 (Yang et al., 2025) as the backbone model. More details are provided in Appendix A.

Evaluation Evaluations are conducted in the MuJoCo simulator across four environments: 2D Locomotion (Hopper), 3D Locomotion (Ant), Swimmer, and Gap Crosser (Gap) (Todorov et al., 2012). Each robot is controlled by a controller learned via an RL method (Yuan et al., 2022), and performance is measured by the distance traveled within a fixed time. Specifically, NGE, ESS, RGS, and our method learn control policies for fixed skeletal structures with PPO (Schulman et al., 2017). In contrast, RL-based methods like T2A and SARD optimize the skeletal structure and control policy simultaneously. To ensure a fair comparison with these baselines, MorphoGen uses reference results from only the first half of T2A’s optimization steps, rather than its fully optimized solutions. In addition to evaluating motion efficiency with an optimal controller, we also wonder how quickly a functional controller can be obtained with limited time and computational resources. To this end, we evaluate each structure using two distinct controllers: a fine-tuned controller (FC), trained for only 0.3M steps, and a fully trained controller (FTC), which is trained until convergence.

5.2 OVERALL PERFORMANCE

To completely demonstrate the quality and diversity of the generated skeleton structures of our methods, we report the top three results, and with name MorphoGenⁱ denotes the top i -th result. We

Method	Fine-tuned Controller (FC)				Fully Trained Controller (FTC)			
	Hopper	Ant	Swimmer	Gap	Hopper	Ant	Swimmer	Gap
ESS	163.4 ± 14.1	108.9 ± 10.1	93.1 ± 11.5	99.6 ± 13.0	658.7 ± 52.7	1387.5 ± 113.3	349.0 ± 28.9	218.6 ± 19.7
RGS	483.5 ± 54.4	508.3 ± 31.0	107.8 ± 13.8	240.7 ± 26.3	849.5 ± 69.0	1394.2 ± 79.0	301.4 ± 31.5	670.8 ± 39.8
NGE	1725.9 ± 115.3	896.2 ± 41.8	119.0 ± 7.0	601.4 ± 40.6	2647.1 ± 224.0	2015.3 ± 205.4	513.9 ± 42.2	822.3 ± 58.0
T2A	2143.2 ± 192.2	1140.2 ± 136.2	703.4 ± 24.6	651.0 ± 76.3	8130.4 ± 1093.8	4268.7 ± 351.4	877.4 ± 65.9	2147.5 ± 322.6
SARD	1841.4 ± 210.4	1174.8 ± 79.6	970.7 ± 17.3	896.3 ± 67.2	7109.0 ± 839.3	4347.1 ± 391.7	892.8 ± 66.6	2532.7 ± 366.2
RoboMorph	/	853.3 ± 60.8	/	/	/	2842.3 ± 353.4	/	/
RoboMoRe	1612.0 ± 193.2	1022.4 ± 87.8	808.1 ± 17.2	718.8 ± 68.8	6697.1 ± 492.4	3514.1 ± 278.9	693.9 ± 34.5	2833.8 ± 286.5
MorphoGen ¹	2474.1 ± 163.6	1560.4 ± 110.3	1056.3 ± 15.5	969.1 ± 62.4	7409.0 ± 654.2	4740.6 ± 282.6	1058.0 ± 37.4	5652.7 ± 280.5
MorphoGen ²	2265.2 ± 135.8	1425.5 ± 71.3	1024.4 ± 15.1	946.7 ± 66.0	7127.9 ± 741.5	4357.9 ± 267.2	1029.4 ± 45.2	5258.3 ± 193.8
MorphoGen ³	2215.0 ± 123.5	1181.3 ± 74.3	1008.6 ± 19.5	860.8 ± 57.7	6384.4 ± 307.2	3994.2 ± 233.4	960.1 ± 17.2	3877.7 ± 141.0

Table 1: Comparison of locomotion score of the generated skeleton structures across different methods, environments and controller. The **bold** values indicate the best-performing solutions, the underlined values denote the second-best solutions, and the *italicized* values represent the third-best solutions (determined separately per column). Each robotic skeleton structure is evaluated using control policies trained across six seeds.

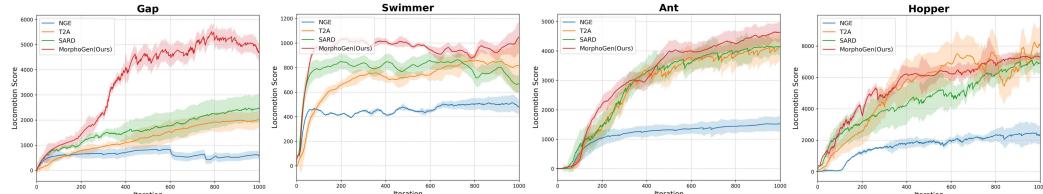
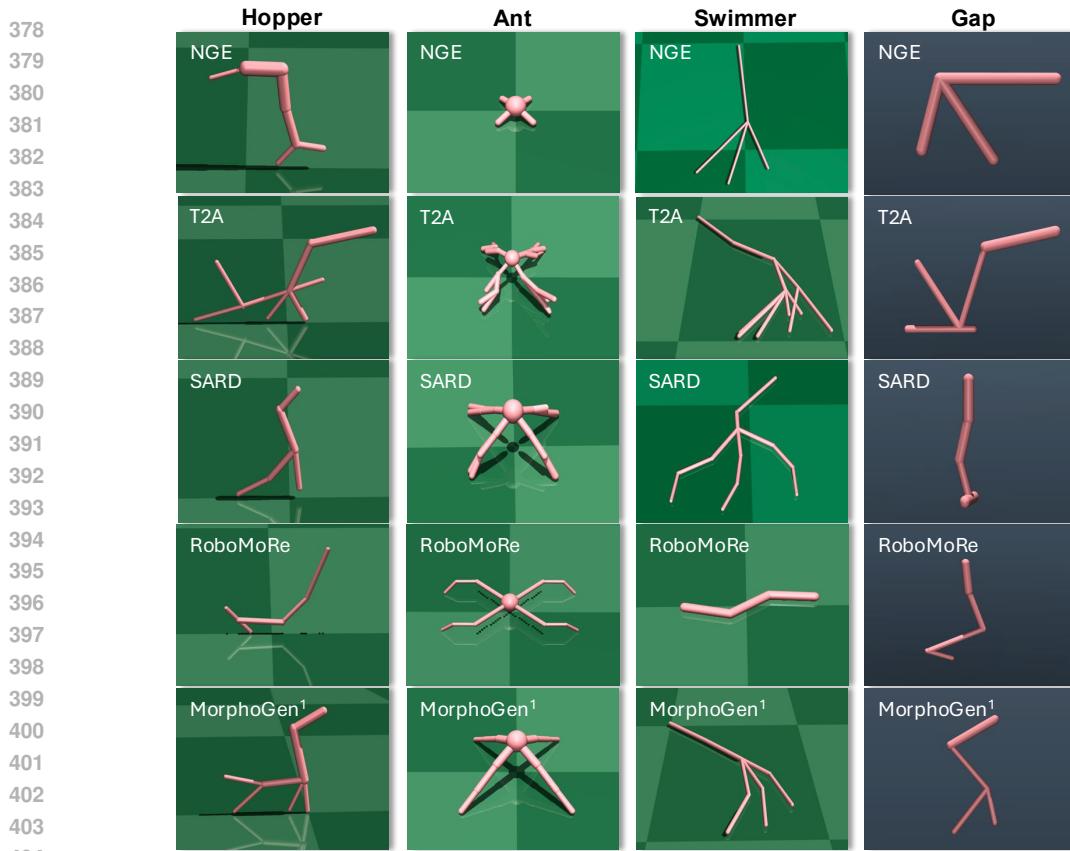


Figure 4: Training curves of locomotion metric, comparing MorphoGen with baseline methods.

summarize the motion performance of the generated robot skeleton structure in Table 1, from which we can have the following observations,

- **Traditional search or evolutionary algorithms fail to provide quality skeleton structures.** The solution space of robot skeletons is extremely large and highly non-convex. Without a clear exploration direction or strong priors, methods such as NGE, ESS, and RGS tend to converge slowly and often produce degenerate structures with poor locomotion ability. These approaches consistently yield the lowest performance, with an average degradation of 76.2% compared to RL-based methods, highlighting the necessity of providing optimizing directions.
- **Constraints on the design space severely limit the quality of generated solutions.** Leading approaches such as T2A, RoboMorph, and RoboMoRe fundamentally restrict the search process by relying on predefined grammar rules, fixed graph topologies, or limited parameter tuning within static skeletons. While these abstractions simplify the optimization problem, they confine the evolutionary search to a narrow, pre-specified subspace, preventing the exploration of the full morphological manifold. As a result, these methods struggle to escape local optima, yielding morphologies with limited locomotion capabilities. Empirically, the average performance of morphologies generated by these constrained baselines is only 63.5% of that achieved by MorphoGen. In contrast, our method operates directly on the raw XML genotype without prior structural assumptions or fixed action spaces. This unconstrained flexibility allows MorphoGen to navigate the immense combinatorial space effectively, unlocking novel, high-performing designs that are inaccessible to methods bound by rigid design priors.
- **Our approach has significant advantages over other methods.** Our method consistently generates skeleton structures that achieve superior locomotion performance across diverse environments. Evaluated by the FTC, our approach attains the best results in the majority of skeletons and ranks within the top two in all cases, yielding an average improvement of 57.5% over state-of-the-art RL baselines. Notably, our method still outperforms all baselines by at least 29.5%, and dominates the top three results in every environment with the FC. These findings indicate that our approach not only enables diverse and high-quality skeleton generation, but also demonstrates strong potential for real-world applications where training resources are constrained.

405 Figure 5: [Visualization of representative robot morphologies generated by different methods.](#)
406413 Figure 6: Top-performing morphologies generated by MorphoGen in the Ant environment.
414

415 To further showcase the efficacy of our approach, we visualize the motion performance of skeleton
416 structures generated by different methods throughout the training process. As shown in Figure
417 4, our method significantly outperforms baseline approaches with the FTC, achieving an average
418 performance improvement of 40.5% across all environments. Moreover, our approach generates
419 skeleton structures that facilitate rapid learning of effective control policies. Specifically, while
420 baseline methods often require hundreds of optimization iterations to achieve acceptable locomotion
421 performance, our method consistently produces skeletons that achieve to effective motion strategies
422 in fewer than 100 iterations.

424 5.3 STRUCTURAL ANALYSIS

426 We provide visual images of skeletal structures generated by different methods, as shown in Figure
427 5. In contrast to the intricate and often incomprehensible structures generated by RL-based
428 methods, the skeletons produced by our approach are clear, intuitive, and have greater potential for
429 real-world implementation. These designs achieve superior motion performance with fewer move-
430 able joints, indicating a higher level of locomotion efficiency. Our method also naturally evolves
431 various bio-inspired structures that resemble real-world organisms. For example, the generated ant
skeleton resembles a spider, the hopper skeleton looks like a horse, and the swimmer skeleton mim-

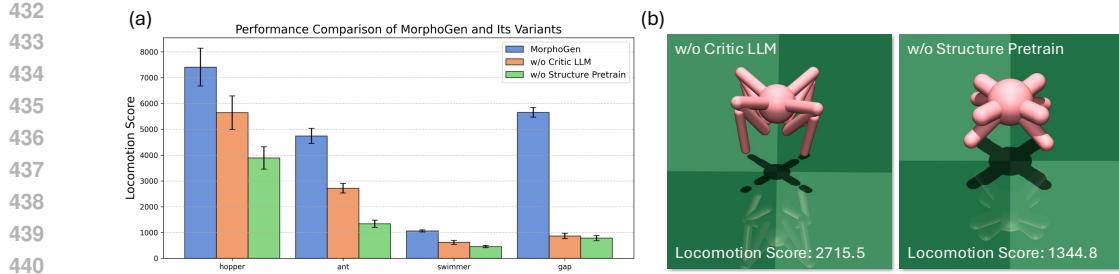


Figure 7: Ablation study on key components of MorphoGen. (a) The performance comparison between MorphoGen and its variants. (b) The corresponding generated robot morphologies.

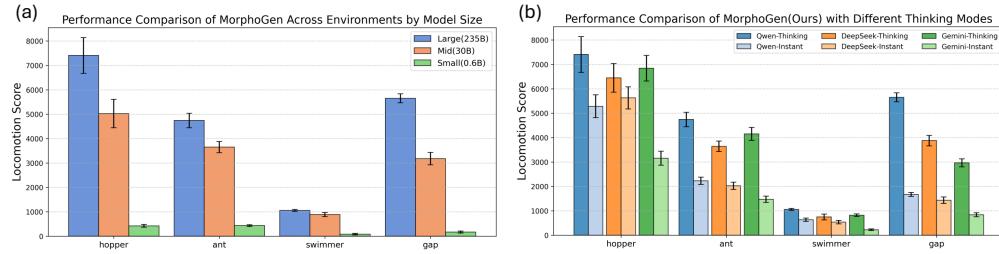


Figure 8: Backbone model selection analysis. (a) The performance of Qwen3 with varying parameter quantities as the backbone model. (b) Performance comparison of models with and without a reasoning component.

ics an octopus. This demonstrates our method’s ability to discover effective and elegant designs by leveraging principles found in nature, which are possibly sourced from the strong priors in LLMs. Furthermore, our approach is capable of generating a diverse range of high-performance skeletons. Figure 6 illustrates the top-performing ant skeletons generated by MorphoGen which achieve high scores while possessing distinct structures and locomotion gaits (see Appendix E for more details). This highlights our method’s capacity to explore a wide variety of effective design solutions rather than converging on a single, narrow set of configurations.

5.4 ABLATION STUDY

To investigate the effectiveness of the employed critic agent and the initial elite solutions, we separately remove these two key components from our framework and report the results of variants in Figure 7. The findings indicate that removing either component significantly degrades the performance of the generated robot skeletons, with the average motion score of the generated structures decreasing by 50.5% and 65.7% respectively. Specifically, the absence of feedback from the critic agent deprives the LLM of a targeted optimization direction during the evolutionary process, severely hindering its exploration efficiency. This is particularly evident in the design of the gap crosser, where the motion score plummets from 5652.7 to 866.4, a significant drop of 84.7%. Meanwhile, without the initial elite solutions, the LLM lacks a crucial set of high-quality examples to learn from. This forces it to waste valuable computational resources on low-quality, sub-optimal designs, as it fails to acquire essential domain knowledge. A prime example is the ant structure, where the motion efficiency of the generated skeletons is only 28.2% of what our full framework achieves. These results underscore the critical role of both the critic agent and the initial elite solutions in enabling our LLM-based framework to generate high-performing skeletal structures.

5.5 BACKBONE MODEL SELECTION

To investigate whether different backbone models influence skeleton design, we conducted experiments focusing on the impact of model size and whether a model is specifically optimized for reasoning. As presented in Figure 8, the size of the backbone model has a significant effect on the performance of the generated skeleton structures. As the model’s parameter scale decreases, the locomotion scores of the produced skeletons drop sharply, from 7409.0 to 424.3(-94.3%), aligning

486 with established scaling laws in LLMs. Furthermore, models specifically designed for reasoning
487 significantly boost the framework’s ability to generate higher-quality motion skeletons. With their
488 enhanced logical and causal reasoning capabilities, are better equipped to understand the intricate
489 relationships between a skeleton’s structure and its physical dynamics. It allows the LLM to more
490 effectively deduce which structural modifications are likely to lead to performance improvements,
491 resulting in more intelligent and efficient evolutionary steps.

492

493 6 DISCUSSION

494

495 In this work, we introduced MorphoGen, an evolutionary framework that reframes the automated
496 design of robot morphologies as a code generation problem. By leveraging LLMs to directly evolve
497 the raw XML genotypes of robots, our approach circumvents the need for intermediate graph-
498 based abstractions and predefined modification actions that seriously constrain the exploration space.
499 Through a combination of a critic LLM providing gradient-like textual guidance , an efficient hier-
500 archical fitness evaluation , and an initial population of high-quality structures , MorphoGen demon-
501 strates a remarkable ability to navigate the vast and complex design space. Our experiments show
502 that MorphoGen not only discovers diverse and high-performing morphologies, but also signifi-
503 cantly outperforms existing baselines across a suite of challenging locomotion tasks.

504 Looking ahead, a promising direction is to enhance the framework’s creative potential by fostering
505 collaboration among multiple, specialized LLMs. Moreover, achieving a co-evolution of morphol-
506 ogy and control policy is also critical. By leveraging the LLM’s understanding of the link between
507 form and function, it could suggest control parameters tailored to the specific morphology. This
508 would create a tighter integration between the evolution of the ”body” and the ”brain,” potentially
509 accelerating the evaluation bottleneck and leading to more holistic and efficient robotic systems.

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540 ETHICS STATEMENT
541

542 This work focuses on the automated design of robot morphologies using LLMs in simulated environments.
543 All evaluations are conducted in physics simulators, ensuring no direct risks to safety,
544 privacy, or the environment. No conflicts of interest or external sponsorships apply, and all methods
545 comply with research integrity standards. We confirm adherence to the ICLR Code of Ethics.

546
547 REPRODUCIBILITY STATEMENT
548

549 We release the complete source code for the MorphoGen framework anonymously, including all
550 scripts required to replicate our experiments. The release will include the set of expert-designed
551 XML files used for structure pretraining, the specific prompts for the coding and critic LLMs, and
552 all hyper-parameters for both the evolutionary search and the control policy optimization.

553
554 REFERENCES
555

556 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-
557 man, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical
558 report. *arXiv preprint arXiv:2303.08774*, 2023.

559 Luzius Brodbeck, Simon Hauser, and Fumiya Iida. Morphological evolution of physical robots
560 through model-free phenotype development. *PloS one*, 10(6):e0128444, 2015.

561 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared
562 Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large
563 language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

564 Nicholas Cheney, Josh Bongard, Vytas SunSpiral, and Hod Lipson. On the difficulty of co-
565 optimizing morphology and control in evolved virtual creatures. In *Artificial life conference*
566 *proceedings*, pp. 226–233. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA
567 journals-info . . . , 2016.

568 Nicholas Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving soft
569 robots with multiple materials and a powerful generative encoding. *ACM SIGEVolution*, 7(1):
570 11–23, 2014.

571 Nicholas Cheney, Josh Bongard, Vytas SunSpiral, and Hod Lipson. Scalable co-optimization of mor-
572 phology and control in embodied machines. *Journal of The Royal Society Interface*, 15(143):
573 20170937, 2018.

574 Rituparna Datta, Shikhar Pradhan, and Bishakh Bhattacharya. Analysis and design optimization of
575 a robotic gripper using multiobjective genetic algorithm. *IEEE Transactions on Systems, Man,*
576 *and Cybernetics: Systems*, 46(1):16–26, 2015.

577 Heng Dong, Junyu Zhang, Tonghan Wang, and Chongjie Zhang. Symmetry-aware robot design with
578 structured subgroups. In *International Conference on Machine Learning*, pp. 8334–8355. PMLR,
579 2023.

580 Jiawei Fang, Yuxuan Sun, Chengtian Ma, Qiuyu Lu, and Lining Yao. Robomore: Llm-based robot
581 co-design via joint optimization of morphology and reward. *arXiv preprint arXiv:2506.00276*,
582 2025.

583 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
584 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
585 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

586 Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. Metamorph: Learning universal controllers
587 with transformers. *arXiv preprint arXiv:2203.11931*, 2022.

588 Zhanpeng He and Matei Ciocarlie. Morph: Design co-optimization with reinforcement learning via
589 a differentiable hardware model proxy. In *2024 IEEE International Conference on Robotics and*
590 *Automation (ICRA)*, pp. 7764–7771. IEEE, 2024.

594 Jiaheng Hu, Julian Whitman, and Howie Choset. GIso: Grammar-guided latent space optimization
 595 for sample-efficient robot design automation. In *Conference on Robot Learning*, pp. 1321–1331.
 596 PMLR, 2023.

597 Yifeng Jiang, Tingnan Zhang, Daniel Ho, Yunfei Bai, C Karen Liu, Sergey Levine, and Jie Tan. Sim-
 598 gan: Hybrid simulator identification for domain adaptation via adversarial reinforcement learn-
 599 ing. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2884–2890.
 600 IEEE, 2021.

601 Yeonju Kim, Zherong Pan, and Kris Hauser. Mo-bbo: Multi-objective bilevel bayesian optimiza-
 602 tion for robot and behavior co-design. In *2021 IEEE International Conference on Robotics and*
 603 *Automation (ICRA)*, pp. 9877–9883. IEEE, 2021.

604 Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information*
 605 *theory*, 37(1):145–151, 2002.

606 Fei Liu, Xi Lin, Shunyu Yao, Zhenkun Wang, Xialiang Tong, Mingxuan Yuan, and Qingfu Zhang.
 607 Large language model for multiobjective evolutionary optimization. In *International Conference*
 608 *on Evolutionary Multi-Criterion Optimization*, pp. 178–191. Springer, 2025.

609 Xingyu Liu, Deepak Pathak, and Kris M Kitani. Herd: Continuous human-to-robot evolution for
 610 learning from human demonstration. *arXiv preprint arXiv:2212.04359*, 2022.

611 Xingyu Liu, Deepak Pathak, and Ding Zhao. Meta-evolve: Continuous robot evolution for one-to-
 612 many policy transfer. *arXiv preprint arXiv:2405.03534*, 2024.

613 Haofei Lu, Zhe Wu, Junliang Xing, Jianshu Li, Ruoyu Li, Zhe Li, and Yuanchun Shi. Bodygen:
 614 Advancing towards efficient embodiment co-design. *arXiv preprint arXiv:2503.00533*, 2025.

615 David Matthews, Andrew Spielberg, Daniela Rus, Sam Kriegman, and Josh Bongard. Efficient auto-
 616 matic design of robots. *Proceedings of the National Academy of Sciences*, 120(41):e2305180120,
 617 2023.

618 Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint*
 619 *arXiv:1504.04909*, 2015.

620 Alexander Novikov, Ng  n V  , Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt
 621 Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian,
 622 et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint*
 623 *arXiv:2506.13131*, 2025.

624 Deepak Pathak, Christopher Lu, Trevor Darrell, Phillip Isola, and Alexei A Efros. Learning to
 625 control self-assembling morphologies: a study of generalization via modularity. *Advances in*
 626 *Neural Information Processing Systems*, 32, 2019.

627 Mateusz Pawlik and Nikolaus Augsten. Tree edit distance: Robust and memory-efficient. *Information*
 628 *Systems*, 56:157–173, 2016.

629 Kevin Qiu, W  dys  w Pa  ucki, Krzysztof Ciebiera, Pawe   Fijalkowski, Marek Cygan, and   ukasz
 630 Kuci  nski. Robomorph: Evolving robot morphology using large language models. *arXiv preprint*
 631 *arXiv:2407.08626*, 2024.

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

634 Asankhaya Sharma. Openevolve: an open-source evolutionary coding agent, 2025. URL <https://github.com/codelion/openevolve>.

635 Karl Sims. Evolving virtual creatures. In *Seminal Graphics Papers: Pushing the Boundaries*,
 636 *Volume 2*, pp. 699–706. 2023.

637 Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applica-
 638 tions: a comprehensive survey. *Artificial Intelligence Review*, 55(2):945–990, 2022.

648 Junru Song, Yang Yang, Huan Xiao, Wei Peng, Wen Yao, and Feifei Wang. Laser: Towards diversi-
 649 fied and generalizable robot design with large language models. In *The Thirteenth International*
 650 *Conference on Learning Representations*, 2025.

651

652 Haoxiang Tian, Xingshuo Han, Guoquan Wu, An Guo, Yuan Zhou, Jie Zhang, Shuo Li, Jun Wei, and
 653 Tianwei Zhang. An ILM-empowered adaptive evolutionary algorithm for multi-component deep
 654 learning systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39,
 655 pp. 20894–20902, 2025.

656 Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control.
 657 In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033.
 658 IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

659

660 Tingwu Wang, Yuhao Zhou, Sanja Fidler, and Jimmy Ba. Neural graph evolution: Towards ef-
 661 ficient automatic robot design. In *7th International Conference on Learning Representations*,
 662 *ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=BkgWHnR5tm>.

663

664 Jie Xu, Andrew Spielberg, Allan Zhao, Daniela Rus, and Wojciech Matusik. Multi-objective graph
 665 heuristic search for terrestrial robot design. In *2021 IEEE international conference on robotics*
 666 *and automation (ICRA)*, pp. 9863–9869. IEEE, 2021.

667

668 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu,
 669 Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint*
 670 *arXiv:2505.09388*, 2025.

671

672 Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun
 673 Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning*
 674 *Representations*, 2023.

675

676 Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris M. Kitani. Transform2act: Learning a
 677 transform-and-control policy for efficient agent design. In *The Tenth International Conference on*
 678 *Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
 679 URL <https://openreview.net/forum?id=UcDUxjPYWSr>.

680

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

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702
703

A BASELINE DETAILS

704
705
We provide additional implementation details for all baselines below.706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

- **Neural Graph Evolution (NGE)** (Wang et al., 2019) formulates robot design as a graph search problem and employs graph neural networks as controllers to share policies across different morphologies. This significantly reduces evaluation cost during search and enables efficient discovery of kinematically meaningful designs.
- **Evolutionary Structure Search (ESS)** (Cheney et al., 2018) jointly evolves morphology and control in soft robots. It introduces *morphological innovation protection*, which temporarily relaxes selection pressure on recently mutated morphologies, allowing controllers to readapt and thus preventing premature convergence.
- **Random Graph Search** (Wang et al., 2019) performs naive random sampling of graph-structured morphologies without guided exploration. Although simple, it serves as a strong baseline to highlight the advantage of structured evolutionary search.
- **Transform2Act** (Yuan et al., 2022) treats robot design as a conditional decision-making process and learns policies that jointly determine both the morphology and control. By leveraging reinforcement learning with modular architectures, it achieves improved sample efficiency over pure evolutionary approaches.
- **Symmetry-Aware Robot Design (SARD)** (Dong et al., 2023) incorporates group-theoretic symmetry constraints into the robot design process. By searching for and enforcing optimal symmetry subgroups, it reduces the design space, improves control stability, and produces more efficient morphologies across diverse tasks.
- **RoboMorph** (Qiu et al., 2024) employs a structured grammar library to describe the sequence of morphology designs. An evolutionary loop retains the highest-fitness individuals as parents for the next generation.
- **RoboMoRe** (Fang et al., 2025) fixes the graph structure of morphology and searches for the best parameters of limbs. It follows a coarse-to-fine paradigm and jointly optimizes morphology and the reward function.

732
733

B CONTROLLER OPTIMIZATION

734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
The controller optimization for each generated robot structure is performed using RL to develop an adaptive control policy π_C for locomotion tasks. The problem is formulated as an infinite-horizon discounted Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $\rho_0 : \mathcal{S} \rightarrow [0, 1]$ is the initial state distribution, and $\gamma \in [0, 1]$ is the discount factor. For a fixed robot morphology, the objective is to learn a control policy $\pi_C : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the expected cumulative discounted reward:

$$J(\pi_C) = \mathbb{E}_{s_0 \sim \rho_0, a_t \sim \pi_C(\cdot | s_t), s_{t+1} \sim P(\cdot | s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right],$$

745
746
where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ are the state and action at time t .747
748
We employ Proximal Policy Optimization (PPO) (Schulman et al., 2017) to optimize the policy, following the approach in Yuan et al. (2022) for locomotion control.

756 C PROMPTS
757758 Coding LLM
759

760 System Message:

761 You are an expert in robotics, MuJoCo simulation, and robot morphology design. Your
762 task is to optimize a robot design defined in MuJoCo XML format to maximize locomotion
763 performance while adhering to best practices.

764 Key Design Principles:

- 765 1. Structural Stability: Ensure a robust base and secure joint connections for balance.
- 766 2. Locomotion Efficiency: Design limbs and joints to maximize forward movement efficiency.
- 767 3. Actuator Coverage: Assign motor actuators to all movable joints.
- 768 4. Physical Realism: Use realistic sizes, masses, and joint ranges for simulation accuracy.
- 769 5. Complexity Balance: Optimize capability without overly complex structures that hinder
770 control.

771 XML Structure Requirements:

- 772 - Position the root body at an appropriate height for stability.
- 773 - Ensure all hinge joints have corresponding motor actuators in the `<actuator>` section.
- 774 - Match joint names exactly between `<joint>` and `<motor>` elements.
- 775 - Define realistic joint ranges for natural movement.
- 776 - Specify capsule geometries with accurate fromto coordinates.
- 777 - Maintain valid MuJoCo XML syntax and structure.

778 Please modify one aspect of the robot design per iteration to **isolate improvements: limbs,
779 joints, or bodies.**

- 780 - Adjust limb lengths and orientations to enhance locomotion efficiency.
- 781 - Strategically add or remove body segments to balance capability and simplicity.
- 782 - Optimize body positions and orientations for improved stability and movement. - Ensure
783 design complexity remains controllable.

784 Output:

- 785 - Do not include explanations, comments, or additional text outside the XML.
- 786 - Return only the complete, valid MuJoCo XML.”

787 User Message:

788 Current Morphology Information:

- 789 - Morphology: {current_xml}
- 790 - Locomotion Score: {current_fitness}
- 791 - Structure Tree: {current_structure_tree}
- 792 - Focus areas: {current_critic_ouput}

793 Sampled Morphology Information:

- 794 - Morphology: {sampled_xml}
- 795 - Locomotion Score: {sampled_fitness}
- 796 - Structure Tree: {sampled_structure_tree}
- 797 - Focus areas: {sampled_critic_ouput}

800 Program Evolution History: {evolution_history}

801 Task: Rewrite the program to improve its Locomotion Score. Provide the complete new
802 xml code.

803 # Your modified xml here

804
805
806
807
808
809

810

Critic LLM

811

812

[t] System Message:

813

814

815

816

You are a robotics expert acting as a critic for an evolutionary algorithm designing a MuJoCo robot. Your task is to provide a specific, actionable suggestion for how to improve the robot's design. Do not give general or too specific advice. Your suggestion should be a concrete change to the XML structure related to an aspect.

817

818

User Message:

819

820

821

822

823

824

825

826

827

828

829

830

831

D SENSITIVITY ANALYSIS

832

833

834

835

836

837

838

839

840

841

Utilizing strong initial genotypes is a widely adopted practice (Qiu et al., 2024; Fang et al., 2025; Yuan et al., 2022) to ensure the search starts within a functionally meaningful region, thereby maximizing optimization efficiency. However, the initial population may cause profound impact to the evolution process and lead to totally different solutions. To investigate to what degree the initial genotypes will influence the final morphology, we conduct an ablation study that use different initial genotypes. Specifically, We employ the morphologies generated by T2A (Yuan et al., 2022) (which is asymmetry with multi-joint and long limbs) and NGE (Wang et al., 2019) (which is symmetric with single-joint and short limbs) and try each possible combinations. Moreover, we also add a 10% random perturbation (rp) to the structure parameters. to showcase the robustness. We summarize both the characteristics and the locomotion scores of the generated morphologies in Table 2.

842

843

Variant	T2A	T2A(rp)	NGE	NGE(rp)	T2A+NGE	T2A(rp)+NGE(rp)
Characteristics	AS	AS	S	S	S	S
	MJ	MJ	SJ	SJ	MJ	MJ
	LL	LL	SL	SL	LL	LL
Locomotion Score	3858.7	3764.2	2478.5	2608.9	4740.6	4415.2

844

845

846

847

Table 2: Ablation study on different initial genotypes, where "rp" denotes random perturbation, "AS" denotes to asymmetry, "S" denotes to symmetric, "MJ" denotes to multi-joint, "SJ" denotes to single-joint, "LL" denotes to long limbs and "MJ" denotes to short limbs.

848

849

850

851

852

853

854

855

856

857

While the structural characteristics influence the final evolutionary direction, experiments with random perturbation show minimal performance drops. This confirms that the search process is robust and effective as long as the initial population contains a diverse set of potentially beneficial locomotion structures. Moreover, the similar characteristics between the genotypes and final solutions confirm that the final evolved designs are not replicas of the initial expert structures. The initial seeds serve only as high-level guidance.

858

859

860

861

To quantitatively analyze this deviation, we compute the Tree Edit Distance (TED) (Pawlik & Augsten, 2016)-based similarity between the best-evolved solutions and their initial expert genotypes, as demonstrated in Table 3.

862

863

The low similarity scores indicate that the generated morphologies are substantially different. They retain only beneficial high-level patterns while evolving highly optimized structural details, validating that MorphoGen performs effective search in the unconstrained XML space.

Environment	Hopper	Ant	Swimmer	Gap
Similarity	54.1%	22.6%	43.7%	35.1%

Table 3: TED-based structural similarity between the best MorphoGen-evolved solutions and the initial genotypes. Lower values indicate more substantial topological innovation.

E DIVERSITY

To rigorously assess the exploration capability of MorphoGen compared to baselines, we employ two distinct metrics: **TED** (Tree Edit Distance) (Pawlik & Augsten, 2016) and **JSD** (Jensen-Shannon Divergence) (Lin, 2002) of length distribution. We measure the pairwise diversity among the top-5 performing morphologies generated by each method in the Ant environment. Specifically, for TED which measures differences in the graph topology, we calculate the pairwise TED between the Kinematic Trees of the morphologies and then normalize and compute the average. For JDS which indicates greater diversity in the parameter space, we first bin the limb lengths of each morphology then calculate the pairwise JSD values of the resulting length distributions and take the average.

Metric	T2A	RoboMoRe	MorphoGen
TED (Structural Diversity)	0.183	0	0.396
JSD (Parameter Diversity)	0.129	0.165	0.248

Table 4: Morphological and Parameter Diversity. Higher values indicate greater diversity.

The results in Table 4 demonstrate that MorphoGen significantly outperforms existing baseline methods. By operating directly on unconstrained XML and utilizing our explicit selection criteria, MorphoGen achieves superior diversity in both morphological graph structure and structural parameter distributions.

F PROXY FITNESS

The primary bottleneck in evolutionary morphology optimization is the extremely high computational cost of training an optimal controller for every candidate design. Our hierarchical proxy fitness strategy is designed to address this challenge by significantly reducing the evaluation time per morphology, thus enabling greater exploration. The time difference between evaluation methods is substantial: a typical proxy fitness calculation spends approximately 2 minutes, whereas training an optimal controller from scratch costs about 3 hours (a 100 \times increase). Using the full controller for fitness calculation is computationally impractical for extensive evolutionary search.

To study the impact of the proxy fitness on overall optimization performance, we compare three variants on a fixed computational budget, reducing the evolutionary iterations for the full-controller variant to make the study feasible.

Variant	Iterations	Total Time	Hopper	Ant	Swimmer
w/o Proxy Fitness (Optimal Controller)	40	$\sim 108h$	5796.0	3862.8	789.7
w/ Proxy Fitness (Limited Iterations)	40	$\sim 1.2h$	5018.3	3194.6	754.8
w/ Proxy Fitness (Full Search Budget)	100	$\sim 4h$	7409.0	4740.6	1058.0

Table 5: Performance Comparison of Proxy Fitness Variants.

While using an optimal controller yields slightly higher quality designs when the number of iterations is artificially fixed at 40, its massive computational overhead makes extensive exploration impossible. In contrast, the proxy fitness allows the algorithm to evaluate orders of magnitude more morphologies within the same time frame. By utilizing the proxy fitness and facilitating a larger number of evolutionary steps, MorphoGen achieves 28.2% better overall performance, while consuming 96% fewer compute resources per iteration. This confirms that the efficiency provided by

918 the hierarchical proxy fitness is essential for effective evolutionary search in the unconstrained XML
919 design space.
920

921 **G USE OF LLMs**
922

923 In addition to their essential role in generating robot morphologies within MorphoGen, LLMs are
924 used only for polishing some paragraphs. The research ideas and designs are entirely conceived by
925 the authors.
926

927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971