

Bridging Language and Physics: Automated Design of Continuum Robots with Large Language Models

Jingyi Chen^{1†}, Mohan Zhang^{2†}, Laura Yao^{2†}, Yingtai Ni¹, Jianmin Ji¹, Jie Peng¹, Song Wang³, and Tianlong Chen^{2*}

¹School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

²Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, USA

³Department of Computer Science, University of Central Florida, Orlando, USA

[†]Equal contribution ^{*}Corresponding author

Abstract—Large language models (LLMs) have recently emerged as a promising tool for automating robot design from high-level specifications, yet they remain ineffective for robots operating under complex physical interactions. This limitation stems from the gap between language-based reasoning and the physical consequences of embodiment, often resulting in designs with low physical validity. In this work, we propose a multi-layered framework, AID-SR, that establishes a closed loop by translating simulator-observed physical states into structured feedback for the LLM designer. Combined with semantic critique, human feedback, and iterative refinement, the framework promotes the generation of physically feasible and functionally meaningful robot designs. We evaluate our approach on tendon-driven continuum robots across a benchmark of 14 tasks spanning reaching, grasping, locomotion, and manipulation. The proposed framework achieves 96.2% rate for passing the simulation feasibility check and by applying a common reinforcement learning training, 26.7% robots can successfully fulfill the corresponding task. We then fabricate three designed robots of AID-SR that successfully complete the task in real-world. These extensive experiments across simulation and real-world environments demonstrate and break the wall of utilizing the LLMs for automated design of continuum robots.

I. INTRODUCTION

The automated design of robotic systems has long been pursued as a way to reduce human effort and to accelerate the exploration of novel mechanical structures [1]–[3]. Traditionally, this process relies heavily on expert intuition and labor-intensive trial-and-error, particularly when robots operate under complex physical interactions such as large deformation, contact, and nonlinear dynamics [4], [5].

Recent progress in large language models (LLMs) has sparked growing interest in using language-driven generative models to automate robot design from high-level, natural language specifications. A number of recent studies have demonstrated that LLMs can generate syntactically correct simulation files, mechanical descriptions, or executable design code [6]–[10]. While these studies demonstrate the potential of LLMs as design tools, most of them focuses on simplified settings or abstract tasks, such as 2-dimensional locomotion. As a result, the generated designs lack clear relevance to real-world manipulation or human-assistive scenarios, where robots must operate in 3-dimensional spaces and interact purposefully with external objects.

A fundamental limitation lies in the mismatch between how LLMs reason and how physical systems behave. LLMs operate

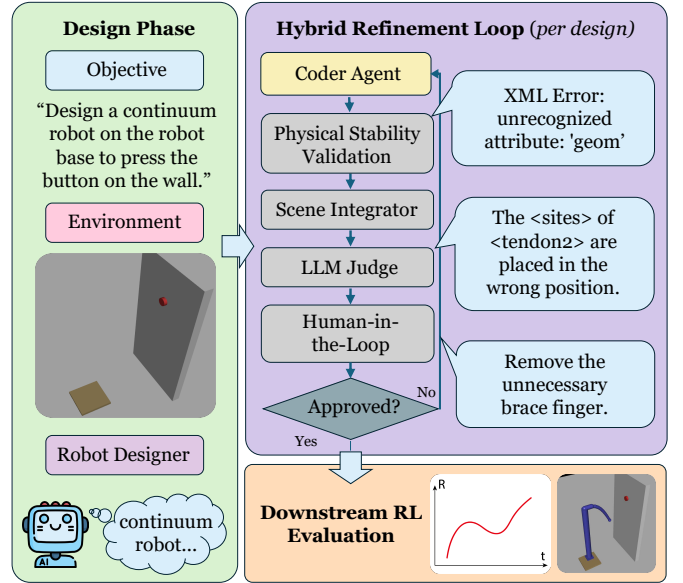


Fig. 1: **The AID-SR Framework Overview.** The process begins with a user-defined objective. The **Designer** agent generates multiple initial design concepts. Each concept enters a parallel **Refinement Loop**, where the **Coder** agent generates a MuJoCo XML file. This design is subjected to four layers of feedback: (1) fast, deterministic **Physical Stability Validation**, (2) specific background scene integration for task functionality, (3) semantic, task-oriented critique from the **LLM Judge**, and (4) high-level guidance from a **Human-in-the-Loop**. The refined design is either approved or sent back to the Coder for another iteration. Approved designs are then evaluated in a downstream RL task.

primarily through linguistic patterns and abstractions, whereas the consequences of physical actions, such as instability, self-collision, or contact failure, are difficult to describe exhaustively in text [11], [12]. As a result, when a generated design fails in simulation, LLMs often lack the grounding needed to understand what went wrong and why. This leads to brittle designs with low physical validity and limited practical relevance, particularly in domains involving soft bodies, contact-rich interactions, and high-dimensional actuation.

In this work, we investigate how LLMs can be systematically

guided to design physically feasible robots for meaningful tasks in 3-Dimensional environments. Rather than relying on one-shot generation, we propose a structured, iterative framework, AID-SR (Autonomous and Iterative Design of Soft Robots), that incrementally exposes the LLM to the physical implications of its own designs. The core idea is to establish a structured feedback pipeline that integrates simulator-observed physical states, semantic critique and human-in-the-loop feedback to guide the LLM designer through iterative robot design refinement. By repeatedly refining robot morphology and physical parameters based on concrete feedback, the LLM is able to progressively improve the physical validity and functional potential of its designs.

We evaluate this framework in the context of tendon-driven continuum robots, a representative class of soft robots that combines high degree-of-freedom with complex contact behavior, while remaining suitable for simulation-based analysis [13]. Tendon-driven continuum robots feature flexible backbones actuated by tensioned tendons, offering rich expressiveness but posing significant challenges for design automation [14]. Their sensitivity to geometric parameters, material properties, and tendon routing makes them a compelling testbed for studying physically grounded robot design with LLMs.

To evaluate our approach, we construct a benchmark of 14 representative tasks spanning reaching, grasping, locomotion, and complex manipulation. For each generated robot design that passes physical validation, we apply a basic reinforcement learning approach. Across the 14 tasks, an average of 26.7% of the robots achieve the success criterion. Finally, to validate real-world feasibility, we fabricate and deploy three LLM-designed robots in physical experiments, demonstrating the contribution to real robot design beyond purely simulated results. The contributions of this paper are threefold:

- We systematically formulate and analyze the problem of **physically grounded robot design with large language models**, identifying the fundamental mismatch between language-based reasoning and physical system behavior, and clarifying the key challenges arising from task-oriented interaction, object contact, and physical embodiment.
- We propose a **multi-layered, physically grounded framework** that incrementally exposes large language models to the physical consequences of their designs, enabling effective and iterative generation of physically valid robot designs. We integrate a lightweight **human-feedback component**, allowing domain experts to efficiently inject tacit engineering knowledge without hand-authoring rules or templates.
- We demonstrate the effectiveness of the proposed framework on the **automated design of tendon-driven continuum robots**, achieving efficient simulation-based design and successful sim-to-real transfer, thereby advancing the practical design of robots with real-world relevance.

II. RELATED WORK

A. Automated Robot Design

Early studies optimize robot designs by tuning parameters of predefined structures or components [15], typically using genetic algorithms [16] or gradient-based methods [17]. These approaches depend on human-specified design spaces, which can either lead to inefficiency when overly broad, or limit expressiveness when overly constrained. With the emergence of generative AI, diffusion-based models have been proposed to learn from robot design datasets and generate robots of similar types [18], [19]. However, their performance is closely tied to the quality and diversity of the training data.

More recently, several works leverage LLMs to directly generate code representations of robot mechanical structures. Some studies focus on idealized 2D simulation environments [6]–[8], where robots are composed of simple blocks with different materials. Others extend this paradigm to 3D simulation [9], [10], but remain limited to relatively simple multi-legged crawling robots. Although some generated designs can be physically fabricated [8], these approaches generally operate within restricted design spaces and narrowly defined tasks. A key limitation is that LLMs have limited ability to perceive and interpret the consequences of their designs within simulation environments, which constrains their applicability to complex models and tasks.

B. Continuum Robots as a Design Challenge

Continuum robots are robotic systems with compliant backbones capable of continuous deformation, offering high adaptability and safety in contact-rich environments [13], [20]. These properties make them suitable for applications such as minimally invasive surgery and manipulation in confined or unstructured spaces [21]. Among various actuation paradigms, tendon-driven continuum robots are prominent due to their compact structure and ability to generate versatile bending and twisting motions through tendon tension control [14].

Many soft robots exhibit large deformations and nonlinear behaviors that challenge conventional modeling methods [22]. For tendon-driven continuum robots, a common approach is to approximate the deformable body with discretized backbone segments connected by joints or elastic elements [23], achieving a practical trade-off between physical fidelity and computational efficiency suitable for simulation and learning-based control.

Despite the structural regularization, tendon-driven segmented-backbone continuum robots retain considerable design flexibility. Simple combinations of rigid segments and tendons have enabled diverse morphologies, including octopus-inspired manipulators [24] and DNA-helix-like structures [25], [26].

From the perspective of automated and language-based design, such robots constitute a challenging yet well-structured testbed. Their design space is amenable to symbolic and parametric representation, while remaining sufficiently rich to capture the high-dimensionality and complex contact behaviors of soft robots.

III. METHOD

In this section, we first formulate the problem of autonomous robot design and highlight the limitations of naive generative approaches III-A. We then provide an overview of our proposed agentic framework, AID-SR, as illustrated in Figure 1. We detail its core design pipeline, which consists of an initial design conception stage and a multi-layer feedback loop enabling iterative refinement in III-B. Next, we further define the design space by specifying the structural and functional constraints of tendon-driven continuum robots, which constitute the target domain of our experiments in III-C. Finally, we describe the evaluation protocol for assessing task completion, using reinforcement learning to verify convergence and task success beyond physical validity in III-D.

A. Problem Formulation

We formally define the task of autonomous robot design as follows: given a natural language high-level objective O (e.g., “design a robot that can crawl through a narrow tunnel”) and an environment specification E in MuJoCo XML format, the goal is to generate a set of valid and functionally promising MuJoCo XML scenes $\{S_1, S_2, \dots, S_k\}$. Each scene S_i must have a physically stable continuum robot morphology R_i , capable of learning a control policy to address the objective O .

To motivate our structured approach, we first investigated a baseline where a state-of-the-art LLM (GPT-5) was directly prompted to generate a complete MuJoCo scene from an objective. We directly provided it with the objective description O and the environment file E , and then required it to generate one complete scene S_1 . This naive method proved ineffective, achieving a success rate of nearly 0% in producing syntactically correct and physically stable files. The predominant failure modes included generating invalid XML syntax, defining physically unstable structures that immediately collapse or explode in simulation, and creating morphologies functionally irrelevant to the task. This result highlights that successful generation requires a structured process of validation and iterative refinement, rather than simple, one-shot translation.

B. The AID-SR Framework: An Agentic Workflow for Soft Robot Design

We introduce our framework, AID-SR (Autonomous and Iterative Design of Soft Robots), depicted in Figure 1. AID-SR is founded on the principle of decomposing the complex design task into specialized roles handled by different agents within a closed-loop system. This agentic structure enables a robust and systematic exploration of the design space.

Initial Design Conception. The process begins with a *Designer* agent that interprets the user’s objective O to propose several distinct, high-level design concepts. This initial step is crucial for promoting design diversity, preventing premature convergence to a single, potentially suboptimal idea. By seeding the process with a range of creative starting points (e.g., for the same locomotion task, they can be single-body peristaltic robots, multi-legged crawling robots, ring-shaped or other unconventional morphologies, etc.), the framework is able to

explore disparate regions of the vast design space in parallel. After the designs are created, we utilize a *Coder* agent to generate an initial version of the robots $\{R_1, R_2, \dots, R_k\}$ based on the original designs.

The Hybrid Feedback and Refinement Loop. This loop is the core of our framework, where each initial design is iteratively improved and validated. It integrates four critical feedback layers:

① *Physical Stability Validation:* This component serves as the key bridge between language-driven design generation and physical feasibility. Each generated robot design R_i is automatically and deterministically validated through direct interaction with the MuJoCo simulator, emulating the diagnostic workflow of a human robotics engineer. The design produced by the *Coder* agent is first loaded into the simulator, where load-time failures such as syntax errors or invalid model definitions are immediately detected and returned to the *Coder* for correction. If instantiation succeeds, the model undergoes a targeted simulation phase in which predefined actuator tests are executed while sensor outputs and simulator states are monitored. Abnormal behaviors including constant sensor readings, high-frequency oscillations, or unexpected simulator resets, are analyzed using predefined diagnostic rules to identify common physical issues, such as ineffective actuation, self-collisions, or numerical instability. Based on these diagnostics, the design is classified as physically stable or unstable, and structured feedback is provided to guide iterative refinement. This automated process replaces manual debugging and enables large-scale, iterative robot design with LLMs.

② *Scene Integration:* In this stage, the *Scene Integrator* agent takes the generated (and correct) robot file R_i , the original environment file E and task objective O as input. Its role is to adapt the robot to the target environment and task by assessing their compatibility and adjusting integration-specific details, such as scaling and placement. The agent then merges the robot and environment descriptions into a unified scene file S_i representing the robot operating within the environment. The resulting scene is subsequently subjected to the same physical stability validation procedure. If instability or simulation errors are detected, the integration process is repeated until a valid scene is obtained. This staged integration strategy decouples robot generation from environment-specific constraints, allowing the model to focus on coherent robot design before task and environment-level adaptation, which empirically leads to more reliable and stable outcomes.

③ *Semantic Critique via LLM Judge:* Designs that successfully pass the integration and physical validation stages are subsequently reviewed by a *Judge* agent, which serves as the final quality gate before human involvement. The Judge assesses whether the complete design file is coherent, well-formed, and sufficiently meaningful to warrant downstream manual check or reinforcement learning, with the explicit goal of avoiding unproductive use of human effort. This evaluation follows a set of predefined semantic criteria, including consistency with the intended design space (e.g., adherence to the definition of a tendon-driven continuum robot), completeness of essential

simulation components such as sensors and actuators, and basic feasibility with respect to the task objective and environment. Based on its assessment of the design file as a whole, the Judge assigns a pass/fail decision. Any designs that fail are accompanied by a structured critique which focuses on which components need improvement and are returned to the earlier agents for revision, while only qualified designs are forwarded for human inspection.

④ *Human-in-the-Loop Refinement*: The framework supports human-in-the-loop refinement after a complete and validated design $\{S_1, S_2, \dots, S_k\}$ has been produced. At this stage, a human engineer may inspect the generated design and provide high-level feedback in natural language, which is then routed back to the initial agent to guide subsequent regeneration. Importantly, this step is not necessary for the framework to function: the automated pipeline can terminate with a final output or continue iterating autonomously without human intervention. However, incorporating human feedback enables the injection of expert judgment and subjective preferences that are difficult to infer automatically. Experimental results demonstrate that such human involvement can improve design quality, highlighting a flexible trade-off between automation and expert labor.

Through iterative execution of the proposed framework, a set of valid and well-formed design files is generated. These qualified designs are subsequently forwarded to downstream reinforcement learning experiments for further evaluation. To ensure computational efficiency and robustness, an upper bound is imposed on the number of framework iterations. If a design repeatedly fails to pass the validation and evaluation stages within this limit, the generation process is terminated, and the corresponding attempt is recorded as an invalid design.

C. Design Space Definition

In this work, we define the design space of tendon-driven continuum robot as follows: a soft structure that leverages under-actuation to achieve passive compliance, characterized by multiple body segments connected via ball or hinge joints and driven by one or more tendons to produce global shape deformation. After providing code examples to help the LLM understand this mechanical unit, we prompt it to autonomously select primitive geometries, assemble them into a robot, and determine suitable joint parameters, actuator layouts, actuator types, and corresponding control parameters. To ensure that the LLM generates syntactically valid XML files during iterative design, we also integrate the MuJoCo modeling specification into the prompt.

During the design generation stage, we provide only a brief task description as the objective O for each scenario, without offering any further hints regarding mechanical configuration or implementation strategies. We instruct the LLM to first articulate its task interpretation and mechanical design rationale, and then produce a complete XML file through the iterative framework loop, which is subsequently fed into the simulation framework for verification and further refinement.

D. Performance Evaluation

To evaluate the functional quality of the robot morphologies generated by our framework, we adopt a standardized downstream evaluation pipeline based on reinforcement learning. The goal of the RL training is not to invest extensive human effort or optimization to achieve the best performance for each robot morphology. Instead, for all robot designs performing the same task, we employ basic and uniform RL training to demonstrate their potential to accomplish the task, thereby validating the effectiveness of the proposed framework.

Reinforcement Learning. Across all tasks, we employ the same reinforcement learning algorithm, Proximal Policy Optimization (PPO) [27], to ensure consistency and comparability. While the learning algorithm is fixed, each task defines its own observation space, action space, reward function, and task-specific success criteria, reflecting the differing objectives and interaction requirements of reaching, grasping, locomotion and manipulation. Detailed specifications of these task formulations are provided in the appendix. For each auto-generated design, the corresponding task configuration is passed to a unified reinforcement learning training pipeline and trained using the same training algorithm, hyperparameters, and number of environment steps. After training, the learned policy is evaluated against predefined task success criteria.

Reward Details. Notably, we avoid morphology-specific reward shaping. Instead, all designs are trained under a *single, generic reward formulation* that provides only sparse, task-level feedback based on progress toward task completion which reflects the practical constraint that the framework generates a large number of structurally diverse continuum robots, for which manual reward engineering would be infeasible. More importantly, it ensures a fair and consistent evaluation across designs, as no individual robot benefits from customized reward signals tailored to its specific morphology or dynamics.

Design Validity. A design is also considered invalid if the robot stays still or produces implausible motion, or the simulator crashes during the training or evaluation process. Designs that complete training without triggering any such failure modes are further confirmed through manual inspection to check if they are structurally plausible continuum robots, and are classified as valid designs. Among valid designs, those that satisfy the task-specific success criteria during evaluation are classified as successful designs. This two-stage evaluation protocol allows us to distinguish failures due to infeasible morphology from those due to control difficulty, and provides a principled measure of the practical utility of LLM-generated robot designs.

IV. EXPERIMENTS

A. Task Design

To evaluate the ability of LLMs to generate functional soft robotic designs, we construct a comprehensive benchmark consisting of diverse task scenarios and systematic experiments (Fig. 2). The benchmark is organized into four major task categories, inspired by prior studies on soft and continuum

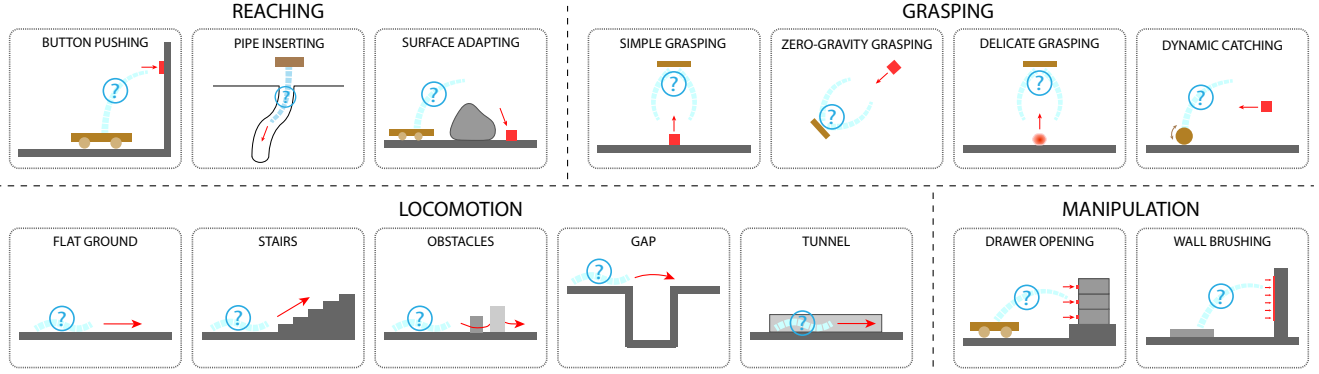


Fig. 2: The test suite consists of 14 specific tasks across 4 categories. In the schematic illustration, the blue components denote the soft robot, whose detailed design is entirely generated by the LLM; the brown components represent a robot base that can move in predefined ways; the gray components indicate fixed structures in the environment; and the red elements denote the task objectives.

robots, and includes a total of fourteen subtasks. Each subtask is instantiated with multiple scenario variations to assess robustness and generalization.

The first category of tasks is *reaching*: controlling the robot’s deformation so that certain functional body parts reach specified target points within specific environments, typically within a workspace comparable to the robot’s own scale [14] [28]. This task evaluates the LLM’s understanding of how a soft robot can achieve purposeful deformation.

The second category is *grasping*: one of the most common tasks in robotics is using a gripper to properly pick up an object, maintain a stable hold, and deliver it to a target location [29] [30] [31] [32] [24] [33]. This task evaluates the LLM’s understanding of how to design one of the most typical soft robot end-effectors. Note that here we do not consider the subsequent placement phase, though it is also a highly valuable direction for future exploration.

The third category is *locomotion*: controlling the robot’s movement through specific environments, so that its entire body travels to a destination far beyond its own body scale [34] [35]. This task examines the LLM’s comprehension of how a soft robot can generate and sustain overall motion.

The fourth category involves *complex manipulations*, where we design two relatively sophisticated tasks to explore the potential of LLMs. The first is drawer opening: the soft robot must repeatedly pull out and push in three drawers arranged vertically [36]. The second is force-controlled brushing: the soft robot must actively brush an entire wall surface while maintaining a controllable contact force [37].

B. Experimental Setup

All agents in our framework are powered by GPT-5, unless specified otherwise in our ablation studies. We set a maximum of 8 refinement iterations for each design concept and fully generate 3 concepts/designs per run. Each LLM call within the framework is limited to a maximum of 20000 tokens, but each call usually consumed fewer than this limit. For each benchmark task, we generate 15 different designs to completion.

To enable practical inspection and iterative steering, we implement a lightweight web interface that exposes the current MuJoCo scene, agent logs, and the generated XML artifact for each design iteration. As shown in Fig. 3, users can (a) configure the task/environment and launch the pipeline, (b) monitor the agent execution and simulation after XML validation, and (c) approve a design or provide text feedback to trigger another refinement round. An end-to-end screen recording is provided in the supplementary video.

The simulation experiments are conducted using the MuJoCo 3.3.4 physics engine [38], wrapped via the Gymnasium 1.1.1 interface [39]. The observation space is defined as a hybrid feature vector that combines two types of information. One part consists of general proprioceptive data, including the robot’s base position, joint angles, and tendon-related measurements. The other part includes expert-designed, task-specific features, such as the relative distance to the target and readings from force sensors. Additionally, the reward functions are customized for each specific task to guide the agent toward the desired behaviors. We employ the deep reinforcement learning algorithms provided by the Stable Baselines3 (SB3) library [40] to address these tasks. Specifically, we utilize the Proximal Policy Optimization (PPO) algorithm [27]. After the LLM produces a batch of soft robot design proposals, we evaluate these generated designs through multiple testing methods.

For *Reaching and Locomotion*, policies are learned directly through basic reinforcement learning via unconstrained exploration in the original action space. In the button-pushing and surface adapting tasks, we design the reward functions to encourage the robot to deform and bring the designated functional part closer to the target position. In the pipe inserting task, the policy is rewarded for moving the robot’s distal end downward while penalizing excessive contact forces against the pipe wall. For locomotion tasks, the rewards are primarily distance-based, guiding the robot toward minimizing its distance to the target position.

For *Grasping*, we observe that many LLM-generated designs employ symmetric finger structures, where each finger is equipped with a dedicated actuator. When these actuators

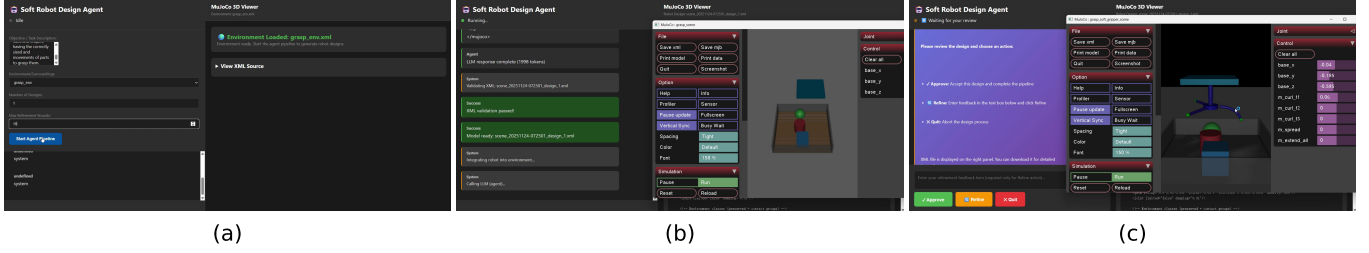


Fig. 3: **Web-based interface for AID-SR.** The UI supports task/environment configuration, live monitoring of the agent pipeline with MuJoCo visualization, and human-in-the-loop decision making (approve/refine/quit). (a) Setup and launch. (b) Generation and simulation monitoring. (c) Review: approve or refine with feedback. With manual text input, each refinement only takes minutes.

are controlled independently, exploration in the raw action space makes it difficult to efficiently discover coordinated grasping behaviors. To improve efficiency, we first examine the design rationale provided by the LLM to identify such symmetric structures. Based on this interpretation, actuators corresponding to symmetric fingers are grouped and controlled jointly, resulting in structured action definitions that better reflect the intended grasping mechanism of each design.

Manipulation involves two relatively complex operation sequences, for which we adopt sequential reward shaping. In the drawer opening task, we sequentially reward the robot for approaching the drawer, executing a grasp, and successfully pulling it open. In the wall brushing task, which requires force control, we reward the robot for following a predefined trajectory to brush the wall while maintaining contact forces within a desired range.

Evaluation Metrics. We use two primary metrics to evaluate our framework’s performance: (1) *Valid Rate*, defined as the percentage of generated designs that are classified as valid after training and evaluation, i.e. designs that complete the simulation without failure and are structurally plausible continuum robots; (2) *Success Rate*, defined as the percentage of generated designs that successfully accomplish the target task during evaluation (success criteria for different tasks are in the Appendix). We report three metrics to evaluate token efficiency: *Average Token Count*, the total number of tokens (prompt + response) consumed to generate a robot design end-to-end; *Average Cost* (\$), the corresponding cost given the token usage and the LLM pricing; and *Average Latency* (s), the wall-clock time required to generate a robot design. For our main results (Table I) and other tables, we focused on the version of AID-SR without the human feedback component (unless otherwise stated) to keep our evaluation consistent and unbiased.

C. Framework Performance on Benchmark Tasks

Based on the experimental results in Table I, the proposed framework can efficiently generate syntactically valid design files; however, due to the inherent limitations of current LLMs, only a small fraction of the generated designs are able to successfully complete the tasks.

Among all tasks, the *Reaching* category exhibits the highest success rate. This is primarily because these tasks require

relatively simple structures: starting from the given robot base, it is generally sufficient to construct a mechanism with adequate workspace and a reasonable range of motion. The *Locomotion* tasks show the next-highest success rate, likely for two reasons. First, these tasks require coherent global motion of the entire structure, imposing higher demands on the overall plausibility of the mechanical design. Second, since a scalable morphology-aware reward scheme for reinforcement learning is not currently feasible, training is unable to fully exploit the potential performance of each generated design. The success rate of the *Grasping* tasks is further reduced. Due to the LLM’s limited understanding of global morphology and physical contact, it rarely produces complete, well-fitted, and structurally stable soft grippers. Finally, the results of the two complex *manipulation* tasks highlight the limitations of current LLMs even more clearly. When faced with sophisticated task sequences and the corresponding mechanical requirements, the model lacks sufficient understanding and generation capability, and therefore produces practically viable designs only with very low probability.

Table I also reports token-efficiency metrics per design, capturing the end-to-end LLM interaction required to produce a valid robot candidate. Across all 14 tasks, our framework uses on average 151K tokens per design, corresponding to an average cost of (\$0.73) per design and 523s latency. Most tasks cluster in a practical band of roughly 100–185K tokens ((\$0.54)–(\$0.88), 411–617s), indicating predictable overhead across reaching, grasping, and locomotion, while more interaction-heavy settings can increase budget (e.g., Force-Controlled Brushing reaches 261K tokens ((\$1.19), 868s)) which corresponds with the framework needing to undergo many more iterations to adequately pass all of the refinement checks. Overall, these results suggest that token usage (and thus cost/latency) scales with task difficulty and constraint complexity, while remaining within a consistent per-design budget for the majority of benchmarks.

D. Ablation Studies and Component Analysis

We perform a series of targeted ablation studies to analyze the proposed framework and quantify the contributions of its key components.

Impact of the Hybrid Feedback System. To isolate the

TABLE I: Main performance of LLM-designed robots across 14 benchmark tasks. These metrics reflect the performance of our framework without the human feedback component, and all token efficiency metrics are reported per design.

Category	Task Name	Valid Rate	Success Rate	Avg. Tokens	Avg. Cost(\$)	Avg. Latency(s)
Reaching	Button Pushing	100%	78.67%	104137	0.57	472.38
	Surface Adapting	100%	13.3%	123515	0.64	499.22
	Pipe Inserting	93.3%	21.4%	212402	0.90	582.97
Grasping	Simple Grasping	93.3%	33.3%	131211	0.65	477.11
	Zero-Gravity Grasping	100%	13.3%	124512	0.62	444.12
	Delicate Grasping	86.7%	20.0%	213250	0.93	602.97
	Dynamic Catching	100%	26.7%	124512	0.62	484.75
Locomotion	Flat Ground	100%	46.7%	103321	0.54	410.92
	Stairs	100%	26.7%	119281	0.60	443.40
	Obstacles	100%	13.3%	143186	0.72	510.39
	Gap	93.3%	53.3%	182576	0.81	530.75
	Tunnel	100%	33.3%	185429	0.88	616.59
Manipulation	Drawer Opening	100%	6.7%	90756	0.48	385.51
	Force-Controlled Brushing	80%	0%	261151	1.19	867.89

effect of each feedback layer, we compare the *Full Framework* against five ablated variants on a subset of representative tasks. These variants include: (1) *Human Feedback*, which performs a single refinement round using only human feedback; (2) *No Judge*, where iterative refinement relies solely on physical validation, without semantic critique; (3) *No Scene Integrator*, in which the environment-aware scene integration component is removed; (4) *No Judge or Scene Integrator*, which disables both components; and (5) *No Validation*, where the Physical Stability Validation module is omitted, allowing designs to bypass MuJoCo-based feasibility checks. As reported in Table II, the full framework achieves the highest overall performance. Removing the Judge results in a 10.7%–20% reduction in task success compared to the full framework without human feedback, indicating that the Coder agent struggles to produce meaningful functional improvements in the absence of structured semantic guidance. Similarly, the *No Scene Integrator* and *No Judge or Scene Integrator* variants exhibit degraded performance, underscoring the importance of environment-aware evaluation and component composition. The *No Human Feedback* setting also leads to a noticeable performance drop, suggesting that human intuition remains beneficial for escaping local optima and aligning designs with nuanced task objectives. Finally, the *No Validation* variant yields a 0% physical validity rate, as none of the generated designs can be reliably instantiated or simulated in MuJoCo, highlighting the indispensable role of physical stability validation in generating viable robot designs.

Analysis of Different Backbone LLMs. We first evaluate the proposed framework using three state-of-the-art backbone LLMs: GPT-5, Gemini 2.5 Pro, and Claude Sonnet 4.5. All three models are able to operate effectively within the framework, with GPT-5 achieving the highest physical validity (100%) and the best overall task performance, and is therefore selected as our default backbone. In our analysis, we prioritize physical validity as the primary comparison metric, as it more directly reflects the model’s ability to identify and repair errors under iterative physical and semantic constraints, whereas task

TABLE II: Ablation study on the hybrid feedback system for the *Flat Ground* and *Button Pushing* tasks. Note that this experiment is conducted separately from the overall experiments reported in Table I. We regenerate a batch of designs for each configuration, except that *Full Framework (w/ human feedback)* is incorporated by adding human evaluations as textual inputs on top of the *Full Framework (w/o human feedback)*, followed by one additional iteration of generation.

Configuration	Valid Rate	Success Rate
<i>Button Pushing Task</i>		
Full Framework (w/o human feedback)	100%	70.7%
Full Framework (w/ human feedback)	100%	76.0%
No Judge	100%	60.00%
No Scene Integrator	86.7%	53.3%
No Judge or Scene Integrator	93.3%	60.00%
No Validation	0%	\
<i>Flat Ground Task</i>		
Full Framework (w/o human feedback)	100%	53.3%
Full Framework (w/ human feedback)	100%	60%
No Judge	80%	33.3%
No Scene Integrator	80%	40%
No Judge or Scene Integrator	73.3%	33.3%
No Validation	0%	\

success exhibits higher variance due to downstream training stochasticity. We also evaluate several recent open-source models and observe a clear degradation in physical validity for smaller-capacity backbones, which frequently fail to recover from invalid designs and remain trapped in repeated infeasible generations. This suggests that sufficient model capacity is important for satisfying the dense constraints and multi-step correction required by iterative robot design.

E. Sim2Real Demonstration and Discussion

In this work, we define the exploration space of the continuum robot and the corresponding task scenarios with physical feasibility in mind, aiming to encourage the LLM to generate designs that are not only functional in simulation but also potentially applicable in real-world settings. To demonstrate this

TABLE III: Performance comparison of different models on the *Flat Ground* task. Valid rate better reflects reasoning and error-correction ability than success rate, while average tokens correlate with both reasoning depth and iterative failures; overall, larger models achieve superior performance.

Model	Valid Rate	Success Rate	Avg. Tokens (per design)
GPT-5	100%	46.7%	103321
Gemini 2.5 Pro	73.3%	0%	113625
Claude Sonnet 4.5	66.7%	26.7%	180761
GPT-OSS-120B	40%	0%	176755
Llama3.3-70B	33.3%	6.67%	117417
Qwen3-30B	0%	\	168070

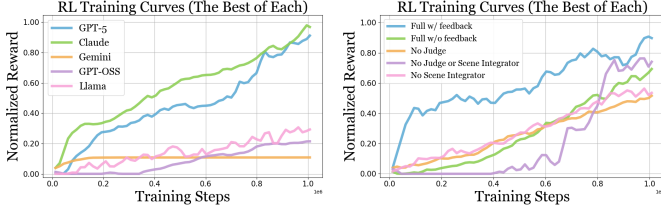


Fig. 4: **Reward curves of the best-performing robots generated by different LLMs or frameworks.** Due to high uncertainty and randomness in individual robot design, differences in reward values do not directly indicate the performance of the LLMs or frameworks themselves; this figure mainly illustrates the effectiveness and stability of the reinforcement learning methods.

feasibility, we select three high-performing designs discovered during simulation and reconstruct them according to their design files. These designs are then physically fabricated using 3D printing and assembled into real robotic systems, which are subsequently evaluated on the same tasks as in simulation.

Since the environment in these three tasks all have large robot base structures, they can accommodate motors as actuators; therefore, the fabrication is relatively straightforward. Qualitative comparisons between simulation and real-world execution are illustrated in Fig.5, where the robots perform identical tasks in both domains. Despite inevitable modeling inaccuracies and unmodeled physical effects, the overall motion patterns, deformation behaviors, and task execution strategies exhibit a high degree of visual and functional consistency.

These observations suggest that the design space defined for the language model possesses a meaningful degree of physical practicality. The ranges of key parameters, such as size, weight, and structural stiffness, largely fall within regimes that can be realized using common fabrication methods and commercially available components. As a result, the generated designs are not merely abstract artifacts of simulation but represent plausible robotic systems that can be instantiated and deployed. This sim-to-real consistency indicates that large language models, when guided by appropriately constrained design representations, can serve as effective tools for proposing physically grounded robot designs with potential practical value.

However, several limitations remain apparent when trying to transfer designs from simulation to the real world. A

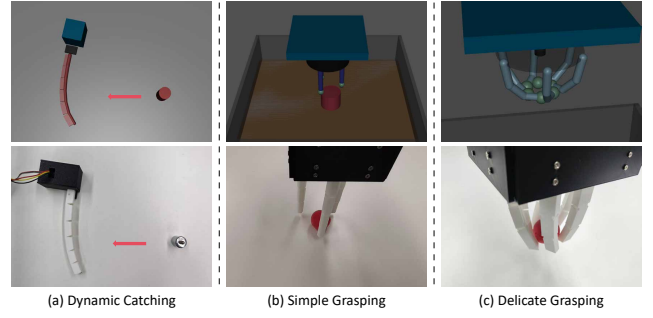


Fig. 5: **Sim-to-real demonstration of three representative designs.** (a) For dynamic catching, the LLM designed a continuum robot segmented into arithmetic lengths. This causes the robot to form a logarithmic spiral, exhibiting grasping capability. (b) For simple grasping, the LLM designed a typical gripper composed of three continuum robots as fingers, each following the same design principle as in (a). (c) For delicate grasping, the LLM proposed an intriguing structure consisting of six fingers, where each joint of every finger bends laterally, forming a helical shape. When all six fingers bend simultaneously, it generates a large contact area, enabling the enclosure of small objects with diverse shapes.

number of designs that perform well in simulation cannot be directly realized in hardware due to idealized modeling assumptions. For instance, tendons in simulation are allowed to ignore intermediate body parts and directly connect distant attachment sites, whereas real cables or ropes must obey geometric constraints and physical routing. Similarly, simulated actuators can assume large gear ratios without accounting for actuator volume, mass, or thermal limits, resulting in power densities that exceed those of nearly all practical actuator technologies. These discrepancies highlight the gap between simulated expressiveness and physical realizability, and point to the need for more restrictive modeling of actuation and transmission mechanisms in future work.

Overall, while a significant sim-to-real gap still exists, our framework succeeds in translating LLM generated designs into fabricated robots that perform meaningful real-world tasks. The successful deployment of three representative designs demonstrates that the LLM can generate robot morphologies with genuine physical competence, rather than designs confined to idealized simulation environments. These results indicate a concrete step toward bridging language-based reasoning and physical embodiment.

V. CONCLUSION

We explore a practical pathway for grounding large language models in physical robot design through a multi-layered, feedback-driven framework. By iteratively incorporating constraints, simulation feedback, and semantic critique, the LLM is able to better interpret the physical consequences of its own designs, resulting in consistently high physical validity. This suggests that such layered interaction is a promising and reusable strategy for LLM-based embodied design. Applied to tendon-driven continuum robots, the framework produces

several designs that successfully transfer from simulation to real-world operation, indicating its potential value as a design aid for more complex robotic systems.

Nevertheless, most of the generated robots tend to be relatively simple, reflecting both the limitations of current simulators and the LLM's still-developing ability to propose unconventional yet reasonable designs. Additionally, the proposed framework can only provide guidance on valid designs, while it lacks reasonable feedback mechanisms regarding the task-level performance. Addressing these limitations remains an important direction for future work.

REFERENCES

- [1] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, no. 6799, pp. 974–978, 2000.
- [2] R. J. Alattas, S. Patel, and T. M. Sobh, "Evolutionary modular robotics: Survey and analysis," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3, pp. 815–828, 2019.
- [3] S. G. R. Prabhu, R. C. Seals, P. J. Kyberd, and J. C. Wetherall, "A survey on evolutionary-aided design in robotics," *Robotica*, vol. 36, no. 12, pp. 1804–1821, 2018.
- [4] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [5] E. W. Hawkes, C. Majidi, and M. T. Tolley, "Hard questions for soft robotics," *Science robotics*, vol. 6, no. 53, p. eabg6049, 2021.
- [6] C. Chen, X. Xu, X. Wang, and X. Huang, "Large language models as natural selector for embodied soft robot design," *arXiv preprint arXiv:2503.02249*, 2025.
- [7] J. Song, Y. Yang, H. Xiao, W. Peng, W. Yao, and F. Wang, "Laser: Towards diversified and generalizable robot design with large language models," in *The Thirteenth International Conference on Learning Representations*, 2025.
- [8] W. Ma, L. Zhao, C.-Y. She, Y. Jiang, A. Sun, B. Zhu, D. Balkcom, and S. Vosoughi, "On the exploration of lm-based soft modular robot design," *arXiv preprint arXiv:2411.00345*, 2024.
- [9] J. Fang, Y. Sun, C. Ma, Q. Lu, and L. Yao, "Robomorph: Llm-based robot co-design via joint optimization of morphology and reward," *arXiv preprint arXiv:2506.00276*, 2025.
- [10] K. Qiu, W. Pałucki, K. Ciebiera, P. Fijałkowski, M. Cygan, and Ł. Kuciński, "Robomorph: Evolving robot morphology using large language models," *arXiv preprint arXiv:2407.08626*, 2024.
- [11] J. Wang, E. Shi, H. Hu, C. Ma, Y. Liu, X. Wang, Y. Yao, X. Liu, B. Ge, and S. Zhang, "Large language models for robotics: Opportunities, challenges, and perspectives," *Journal of Automation and Intelligence*, vol. 4, no. 1, pp. 52–64, 2025.
- [12] A. Chierian, R. Corcodel, S. Jain, and D. Romeres, "Llmphy: Complex physical reasoning using large language models and world models," *arXiv preprint arXiv:2411.08027*, 2024.
- [13] M. Russo, S. M. H. Sadati, X. Dong, A. Mohammad, I. D. Walker, C. Bergeles, K. Xu, and D. A. Axinte, "Continuum robots: An overview," *Advanced Intelligent Systems*, vol. 5, no. 5, p. 2200367, 2023.
- [14] P. Rao, Q. Peyron, S. Lilge, and J. Burgner-Kahrs, "How to model tendon-driven continuum robots and benchmark modelling performance," *Frontiers in Robotics and AI*, vol. 7, p. 630245, 2021.
- [15] W. Li, Z. Wang, R. Mai, P. Ren, Q. Zhang, Y. Zhou, N. Xu, J. Zhuang, B. Xin, L. Gao *et al.*, "Modular design automation of the morphologies, controllers, and vision systems for intelligent robots: a survey," *Visual Intelligence*, vol. 1, no. 1, p. 2, 2023.
- [16] L. Strgar, D. Matthews, T. Hummer, and S. Kriegman, "Evolution and learning in differentiable robots," *arXiv preprint arXiv:2405.14712*, 2024.
- [17] D. Matthews, A. Spielberg, D. Rus, S. Kriegman, and J. Bongard, "Efficient automatic design of robots," *Proceedings of the National Academy of Sciences*, vol. 120, no. 41, p. e2305180120, 2023.
- [18] W. K. Chan, P. Wang, and R. C.-H. Yeow, "Creation of novel soft robot designs using generative ai," *arXiv preprint arXiv:2405.01824*, 2024.
- [19] T.-H. J. Wang, J. Zheng, P. Ma, Y. Du, B. Kim, A. Spielberg, J. Tenenbaum, C. Gan, and D. Rus, "Diffusebot: Breeding soft robots with physics-augmented generative diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 398–44 423, 2023.
- [20] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE transactions on robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.
- [21] S. Kolachalama and S. Lakshmanan, "Continuum robots for manipulation applications: A survey," *Journal of Robotics*, vol. 2020, no. 1, p. 4187048, 2020.
- [22] L. Qin, H. Peng, X. Huang, M. Liu, and W. Huang, "Modeling and simulation of dynamics in soft robotics: A review of numerical approaches," *Current Robotics Reports*, vol. 5, no. 1, pp. 1–13, 2024.
- [23] A. Bajo and N. Simaan, "Kinematics-based detection and localization of contacts along multisegment continuum robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 291–302, 2011.
- [24] Z. Wang, N. M. Freris, and X. Wei, "Spirobs: Logarithmic spiral-shaped robots for versatile grasping across scales," *Device*, vol. 3, no. 4, 2025.
- [25] W. Hu and G. Alici, "Bioinspired three-dimensional-printed helical soft pneumatic actuators and their characterization," *Soft robotics*, vol. 7, no. 3, pp. 267–282, 2020.
- [26] L. H. Blumenschein, N. S. Usevitch, B. H. Do, E. W. Hawkes, and A. M. Okamura, "Helical actuation on a soft inflated robot body," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 245–252.
- [27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [28] L. Mao, P. Yang, C. Tian, X. Shen, F. Wang, H. Zhang, X. Meng, and H. Xie, "Magnetic steering continuum robot for transluminal procedures with programmable shape and functionalities," *Nature communications*, vol. 15, no. 1, p. 3759, 2024.
- [29] H. C. Lee, N. Elder, M. Leal, S. Stantial, E. Vergara Martinez, S. Jos, H. Cho, and S. Russo, "A fabrication strategy for millimeter-scale, self-sensing soft-rigid hybrid robots," *Nature Communications*, vol. 15, no. 1, p. 8456, 2024.
- [30] J. Fishman, S. Ubellacker, N. Hughes, and L. Carlone, "Dynamic grasping with a soft" drone: From theory to practice," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4214–4221.
- [31] K. Fu, X. Wu, S. Yu, Q. Zhang, Y. Yang, T. Xu, W. Yao, T. Yi, H. Li, Y. Chen *et al.*, "Origami exoskeletons for enhanced soft robotic manipulation," *Science Advances*, vol. 11, no. 31, p. eadv6629, 2025.
- [32] Y. Zhang, P. Li, J. Quan, L. Li, G. Zhang, and D. Zhou, "Progress, challenges, and prospects of soft robotics for space applications," *Advanced Intelligent Systems*, vol. 5, no. 3, p. 2200071, 2023.
- [33] N. R. Sinatra, C. B. Teeple, D. M. Vogt, K. K. Parker, D. F. Gruber, and R. J. Wood, "Ultragentle manipulation of delicate structures using a soft robotic gripper," *Science Robotics*, vol. 4, no. 33, p. eaax5425, 2019.
- [34] M. Li, G. Wang, J. Wang, Y. Zheng, and X. Jiao, "Development of an inchworm-like soft pipe robot for detection," *International Journal of Mechanical Sciences*, vol. 253, p. 108392, 2023.
- [35] R. Feng, Y. He, S. Feng, and S. Li, "Impulsive actuation for soft robots," *npj Robotics*, vol. 3, no. 1, p. 27, 2025.
- [36] H. Jiang, Z. Wang, Y. Jin, X. Chen, P. Li, Y. Gan, S. Lin, and X. Chen, "Hierarchical control of soft manipulators towards unstructured interactions," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 411–434, 2021.
- [37] X. Chen, J. Shi, H. Wurdemann, and T. G. Thuruthel, "Vision-based tip force estimation on a soft continuum robot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 7621–7627.
- [38] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [39] S. Schulhoff, J. J. Tai, H. J. S. Tan, O. Younis *et al.*, "Gymnasium: A standard interface for reinforcement learning environments," 2025. [Online]. Available: <https://doi.org/10.5281/zenodo.15753105>
- [40] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of machine learning research*, vol. 22, no. 268, pp. 1–8, 2021.