On the Exploration of LM-Based Soft Modular Robot Design

Weicheng Ma[2,*], Luyang Zhao[1,*], Chun-Yi She[1,*], Yitao Jiang[1], Alan Sun[1],
Bo Zhu[2], Devin Balkcom[1], Soroush Vosoughi[1]

Recent large language models (LLMs) have demonstrated promising capabilities in modeling real-world knowledge and enhancing knowledge-based generation tasks. In this paper, we further explore the potential of using LLMs to aid in the design of soft modular robots, taking into account both user instructions and physical laws, to reduce the reliance on extensive trial-and-error experiments typically needed to achieve robot designs that meet specific structural or task requirements. Specifically, we formulate the robot design process as a sequence generation task and find that LLMs are able to capture key requirements expressed in natural language and reflect them in the construction sequences of robots. To simplify, rather than conducting real-world experiments to assess design quality, we utilize a simulation tool to provide feedback to the generative model, allowing for iterative improvements without requiring extensive human annotations. Furthermore, we introduce five evaluation metrics to assess the quality of robot designs from multiple angles including task completion and adherence to instructions, supporting an automatic evaluation process. Our model performs well in evaluations for designing soft modular robots with uni- and bi-directional locomotion and stair-descending capabilities, highlighting the potential of using natural language and LLMs for robot design. However, we also observe certain limitations that suggest areas for further improvement.

I. INTRODUCTION

Robots constructed from soft, modular components offer the flexibility to be quickly redesigned or reassembled to adapt to new environments [1, 2]. However, designing these robots remains a challenge within the robotics community due to the interleaving complexities involved in the combinatorial exploration of modular compositions and the differentiable optimization of control policies. Adapting these robots to various physical environments, characterized by differing terrain, obstacles, and frictional properties, further exacerbates these challenges. Traditional design pipelines rely heavily on extensive trial-and-error experiments to identify viable designs from a vast pool of possibilities, making the design process particularly tedious when tackling robots consisting of many modules in a complex physical environment [3]. Producing a reasonably optimized design remains impractical, particularly for common users, due to the significant engineering and physics expertise required in the design process.
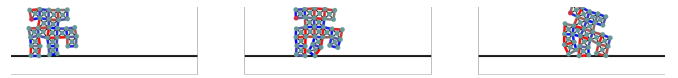

Fig. 1. A robot designed by our model with 2.5 feet in contact with the ground

While a few recent works have begun to explore ML-based models for robot design [4, 5, 6], they often simplify the task by limiting the number of unit robots and predefining the shapes and functionalities of building blocks. These predefined elements rely heavily on expert knowledge in robotics, limiting the flexibility and scalability of these models. To our knowledge, none of the modular robot design frameworks have utilized large language models (LLMs), and we are particularly interested in exploring the potential and capabilities of LLMs for this purpose. In view of recent advances in the field of natural language processing (NLP) where natural language is used to represent task settings and environments in high-dimensional semantic space [7, 8], this paper explores a new direction by exploring the feasibility of using NLP models to generate robot designs in general, unconstrained scenarios. This end-to-end setting adopts natural language as the connecting cord throughout the robot designing pipeline, relieving the requirement of expert-level knowledge and introducing high levels of flexibility to the designs.
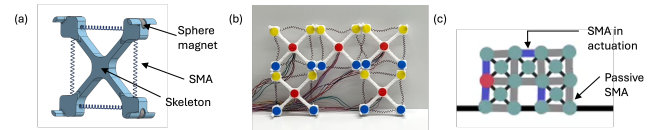

Fig. 2. Module design. (a) A single module is composed of a flexible skeleton, four SMA actuators, and four sphere magnets. (b) An example of real robots built with five modules. (c) A 5-module robot in simulation, where blue lines indicate the springs being actuated, and grey lines indicate the passive SMAs.

Specifically, we investigate how the expressive flexibility of natural language can represent robot designs, enabling the training of large language models (LLMs) for the robot design task. To overcome the difficulty of supervising the

---

training of ML-based robot generation models, we prepare the training data leveraging automatic data augmentation approaches and use a differentiable physics simulation tool to generate control sequences of the designed robots for the target goals. Coupled with the simulator, we propose five simulation-based evaluation metrics for automating the evaluation of robot designs in complex physical environments. These metrics assess the optimality of robot-designing models primarily from the aspects of (1) accommodating the structural requirements in the instructions, (2) producing robots that excel in accomplishing the target task, and (3) designing unseen robots instead of memorizing configurations in the models' training data. It is critical to note that the choice of the physics simulation tool is arbitrary, and we adopted a differentiable physics simulation tool to reduce the need for human labor in designing the control sequences for the generated robots. We then conducted different experiments with our LLM-centered robot-designing framework to show the practicability of using natural language to represent and guide the design of soft modular robots.

Our proposed model is trained using 3,000 randomly selected robot configurations and prompted to generate robot designs given specific environments, target goals, and suggestions. We then match the robots designed by our model with real robots to validate the generations and learn better empirical designs from high-quality designs. To enable such matching, we designed a 2D real robot that is both soft and modular, as depicted in Figure 2(a), to form the generated robot design. Figure 2(b)(c) displays examples of the real robot and its simulated counterpart. From our primary experiments, we discovered multiple generations that differ from traditional human-designed robots but perform well toward various objectives in simulations. For instance, our model designed a robot with two long legs and a short limb (illustrated in Figure 7a) to achieve forward locomotion, the design of which is not intuitive. However, this design locomotes efficiently on the flat plane, benefiting from the short limb alternatively serving as a leg or an arm depending on the current deformation state (Figure 1). Such new design is a departure from the traditional three-legged robot designs and motivates domain experts to come up with less straightforward but effective designs.

II. RELATED WORK

Modular Self-reconfigurable Robots (MSRRs): MSRRs present a significant departure from traditional construction methods. These robots are made up of multiple modules, each possessing the ability to move and mechanically connect with one another[9, 10]. This enables them to restructure and adapt to a variety of tasks [11]. Historically, these modules have been rigid. However, the introduction of compliance and flexibility to these modules has expanded the capabilities of soft modular robots. This evolution offers new modes of actuation, mobility, and even safer interactions with humans [1, 3, 12, 13].

Soft Modular Robots: These robots, with their increased flexibility, can perform a variety of functions by deforming into different shapes from various initial configurations [1, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. For instance, Zhao et al. [24] developed identical self-reconfigurable blocks called Starblocks, which can achieve diverse forms of locomotion by assembling into configurations such as a self-assembled wheel or a quadruped. Additionally, these blocks can be configured into a robotic arm with a gripper for prehensile manipulation, a lattice for non-prehensile manipulation, or even a tent-like structure for formation tasks. Configurations have spanned from simple chain-like formations to sophisticated designs, all aiming to support activities like locomotion, manipulation, and transformation.

Robot Design through Manual Assembly and Iterative Testing: In both rigid and soft self-reconfigurable modular robots, the majority of contemporary research still necessitates the manual design of distinct assembled configurations to fulfill various tasks. Though these methodologies prove effective, they often involve significant labor and multiple iterations of trial and error to pinpoint the optimal configuration [25, 24, 17]. Automating the design of these configurations for various environments and tasks is a crucial area for exploration.

Robot Design Using Learning-Based Techniques: There are several studies that using learning-based methods to design rigid modular robots with different types of components. Whitman et al. [4] apply deep reinforcement learning to design modular serial manipulators for specific tasks. Specifically, the study uses 11 types of modular components, including three base mount orientations, one actuated joint, six different links/brackets, and one end-effector. There are constraints between those modular components, such as the maximum number of modules allowed in a configuration, limited to 16, and the need to ensure physical and functional compatibility between the components, which dictates how they can be arranged to achieve the desired task performance. RoboGrammar, done by Zhao et al., introduce an automated framework that generates optimized robot structures by leveraging a recursive graph grammar and various robot components, including body segments, limbs, and joints, to navigate diverse terrains effectively [5]. Hu et al. [6] investigate robot design using a generative adversarial network (GAN) called RoboGAN, which works with modular robots composed of different types of modules: body, legs, wheels. RoboGAN learns a mapping from tasks (e.g., terrain types) to a distribution of modular robot designs. Unlike traditional methods that typically produce a single optimal design, RoboGAN can generate multiple distinct designs that are all viable for the given task.

However, all related works focus on designs composed of rigid, function-specific modules, which offer relatively lower flexibility compared to soft modular robots. To our knowledge, there is no research on designing soft modular robots. Our work addresses this gap by focusing on identical soft modules that can connect freely in any direction, supporting diverse configurations for various tasks. This design space is vastly larger and not predefined, requiring a generative model to manage its complexity. Additionally,
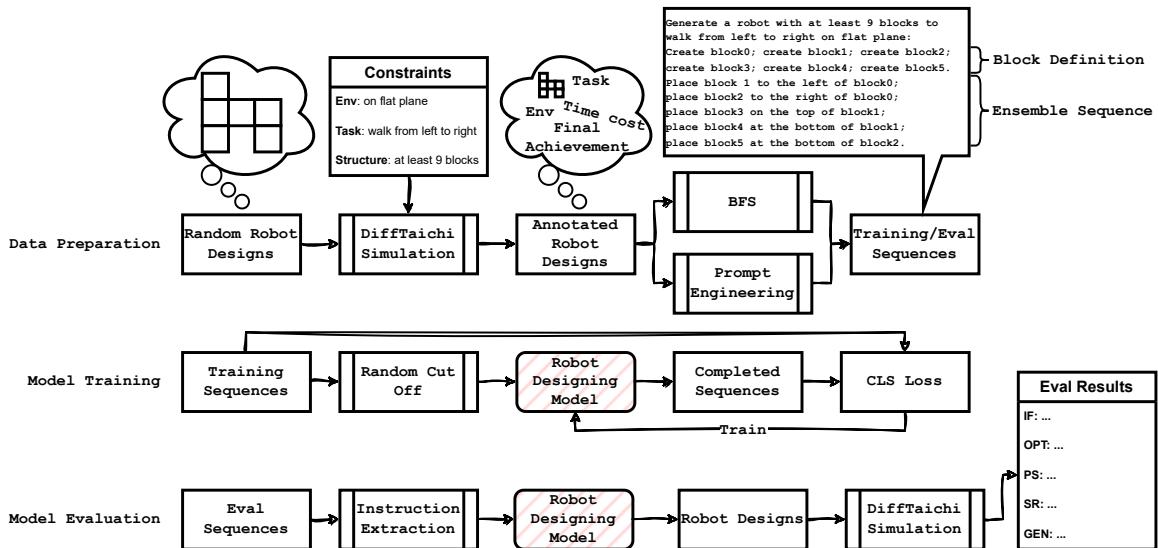
Fig. 3. An overview of the data preparation and model training and evaluation pipelines.

the use of Large Language Models (LLMs) in robot design remains unexplored. Our research aims to investigate LLMs for designing soft modular robots, offering a new direction for further study.

Differential-based Simulation Tools: The use of differential-based simulation tools, such as diffTaichi, has become commonplace in robot gait generation. These tools offer a virtual environment where various robot configurations can be tested for their efficacy, eliminating the need for resource-intensive physical prototypes [26]. Simulation tools have been particularly instrumental in advancing soft modular robot research by offering insights into potential design adaptations before actual assembly.

III. METHODS

This paper explores the use of LLMs for designing soft modular robots, employing a natural language generation input and output style. The following subsections provide details on the configuration and training/evaluation processes of our proposed model.

A. Robot Designing Model

We adopt a generative LLM to design robots with pre-specified environments, task objectives, and structural requirements. Input and output formats of the model are specified in Sections III-A.1-III-A.2, respectively.

1) Environment and Task Representation: Leveraging the LLMs' proficiency in understanding natural language, we craft the prompts in natural language, instructing the model to *"Design a soft modular robot for [task-objective] for [distance] in [environment] using [structural-constraints]."* Here, *[task objective]* and *[environment]* are essential parameters that link the model to the simulation tool, enabling the model to learn to generate legal and high-quality robot designs. *[Distance]* and *[structural constraints]* are optional parameters introducing variability into the prompts and aid in minimizing the risk of overfitting. Below are two example prompts that share the same task objective and environment

but vary in their specifications:

"Design a soft modular robot for walking from left to right on a flat plane using at most 9 blocks."

"Design a soft modular robot for walking from left to right for at least 6 body_length on a flat plane."

To demonstrate the workflow, we design three specific task objectives: unidirectional and back-and-forth locomotion on a horizontal plane, and stair-descending locomotion.

2) Design Representation: The design representation consists of two parts, i.e., a block-definition section and an ensemble sequence section. Specifically, the block definition section specifies the properties and constraints of the individual blocks that will be used in the robot's construction. The robot ensemble section then provides a step-by-step description of how these blocks are assembled to form the complete robot, detailing the spatial and functional relationships between the blocks to achieve the desired design. Since each robot design could be assembled in arbitrary orders, it could be mapped to multiple language representations, enabling easy data augmentation via breadth-first search (BFS). An example design of a robot and one possible language representation of it is illustrated in Figure 3.

B. Evaluation Metrics and Settings

We define five reference-less metrics to evaluate soft modular robot designing models from the instruction-following, optimality, and novelty aspects. **(1) The instruction-following metric (IF)** assesses how closely a model's generated robot designs adhere to the structural requirements specified in the prompts. **(2) The promise score (PS)** estimates the total locomotion distance that each robot designed by the model can achieve within a long duration. **(3) The task optimality metric (OPT)** gauges the efficiency of the robot designs by measuring the time needed to complete the tasks specified in the prompts. **(4) The generalizability metric (GEN)** calculates the percentage of robot designs not previously seen in the model's training data, highlighting the model's ability to improvise instead of memorize. **(5) The**

success rate (SR) measures how frequently the model can successfully generate a legal robot configuration description.

Note that in this study, the capabilities of our framework are constrained by the simulation tool, DiffTaichi, limiting our testing to relatively simple locomotion tasks. Consequently, we tailored the OPT metric to these tasks and performed calculations based on the time it takes for a robot to complete a given task. As we expand our framework to incorporate more advanced simulators and diverse task types, the design of the OPT metric will also be adapted to align more closely with the specific objectives of these new tasks.

C. Training Data Preparation

The training of our model does not rely on human-annotated data but instead utilizes synthesized data generated through DiffTaichi. Specifically, we engineer task objectives and environments, randomize parameters such as the distance between the starting point and the destination or stairs, and simulate movements across various robot configurations. After these simulations, the generated data points are formatted as tuples of (task objective, environment, maximum distance, robot configuration, and time cost) to form the training dataset. The robot configurations are natural language descriptions of robots, expressed using the format exemplified in Figure 3 and diversified using BFS. In our principal experiments, our training data involves 3,000 randomly sampled robot configurations within 5×5 grids.

Leveraging the training data, we devise two categories of natural language prompts for training our model, targeting causal language modeling (CLM) [27] and binary decision-making objectives.

1) CLM Objective: For the CLM objective, data point entries are incorporated into a predefined sentence template, i.e., *"Design a soft modular robot to achieve [task-objective] over a distance of [min/max distance] within [environment] using [min/max number-of-blocks] blocks."* To enhance the diversity of the training prompts and minimize the risk of overfitting, the inclusion of the *distance* and *number-of-blocks* parameters is randomized.

2) Decision-making Objective: For the binary decision-making objective, we randomly pair data points designed for the same environment and task objective (designated as [data1] and [data2]) to create comparative prompts structured as follows: *"For achieving [task-objective] over a distance of [min/max distance] within [environment] using [min/max number-of-blocks] blocks, which design is better? (a) [robot configuration of data1]. (b) [robot configuration of data2]. [robot configuration of winner]."* Here, the "winner" is determined as the configuration with the lower time cost. Similar to the CLM objective, the inclusion of the *distance* and *number-of-blocks* parameters is made optional.

IV. ROBOT DESIGN AND CALIBRATION OF SIMULATION

This section outlines the design of our two-dimensional (2D) robotic model and the subsequent calibration of its simulation counterparts. The calibration process is driven by the utilization of empirical data from real-world robot operations, ensuring that the simulations accurately reflect the behaviors observed in the actual robots.
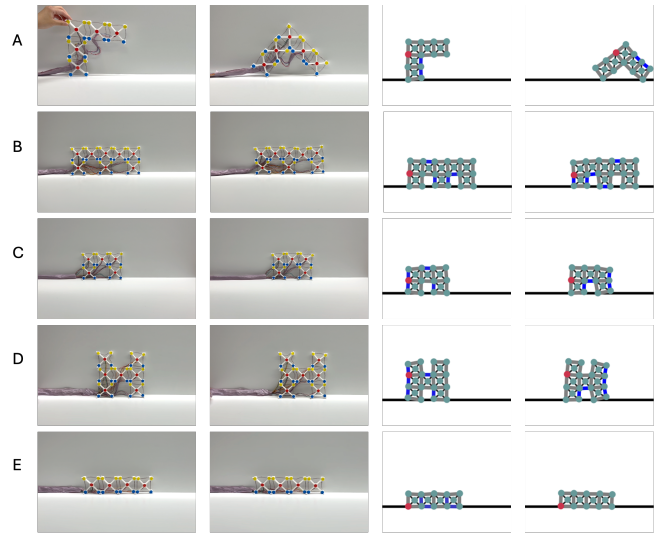
A. Robot design

Fig. 4. Examples of locomotion for five different robot configurations in both the physical world and simulation. For each row, from left to right: initial configuration in the real world, final configuration after four rounds of gait, initial configuration in the simulation, and final configuration in the simulation.

An individual module (Figure 2(a)) consists of a skeleton, four shape memory alloy (SMA) coils (coil diameter: 3.45 mm; wire diameter: 0.51 mm, Dynalloy), and spherical magnets (diameter: 3/8"). The skeleton has an 'x' shape that matches the shape used in the simulator, with four equal-length bars connected at the center. It is printed from flexible TPU material with an infill density of 100% to provide a stronger adversarial bending force, helping restore the module's original shape after SMA contraction. The SMAs are mounted between the adjacent bars.

Each SMA spring is crimped with fishing wire and electric wire in a ferrule and attached to grooves in the skeleton. The detwinned Martensite rest length of each SMA is longer than the distance between the grooves, allowing extension when the skeleton deforms due to SMA actuation. The edge length of each module is about 7.2 cm. The Austenitic (actuated) rest length of the actuators is 1.2 cm, while the detwinned Martensite rest lengths are set to 3.7 cm upon installation. These lengths can change with actuation cycles due to variations in antagonistic forces from other SMA actuators and the elastic energy stored in the deformed soft skeleton. When the SMAs are actuated, they deform the skeleton. As they cool, the restoring force of the skeleton stretches the SMAs back to their detwinned Martensite rest length. We control the SMAs in an open-loop manner by actuating them for a specific duration under 5V. Physical experiments were conducted for each application to determine the appropriate actuation duration. Each SMA coil needs 15-20 seconds to cool down and recover.

Spherical magnets are attached to the ends of each limb of modules to allow them to connect with each other. The polarity of these magnets was set according to the method outlined in [28], where four magnets form a saddle configuration. A saddle of magnets is a polygon with alternating polarities, where magnets point alternately inward and outward. By defining the top and bottom directions of each robot and aligning the magnetic spheres accordingly, any module can be connected from any direction.

B. Calibration of Simulation

To ensure the simulation reflects the behaviors observed in real robots, we manually designed five distinct configurations (shown in Figure 4: a linear chain, a 5-robot two-legged configuration, a 7-robot two-legged configuration, an 8-module three-legged configuration, and a 5-module L-shape configuration. These configurations were chosen to explore the effects of different aspect ratios, the number of legs, and symmetry on locomotion. The L-shape configuration, in particular, was selected because our initial simulations with DiffTaichi and LLM learning showed a preference for configurations that could initially topple due to an unbalanced center of mass. We aimed to investigate how such configurations perform in the real world.

For each configuration, we manually designed the gaits for locomotion (i.e., the control sequence of SMA actuations, detailed control signal can be found in Fig. 5). We applied the same gaits and actuate the same round of gaits to both physical robots and simulations and used the distance traveled relative to the body length of the module as the calibration parameter. Our results show that with parameter fine-tuning in the simulation, the real-world robot and simulation exhibited the same ranking among these five configurations as shown in Fig. 4. The ranking, in both the real robot and simulation, was as follows: 5-module L-shape > 8-module three-legged configuration ≈ 5-robot two-legged configuration > 7-robot two-legged configuration > linear chain. We also noticed that while the L-shape ranked first in both cases, it moved faster in simulation due to inertia and a bouncier skeleton, which doesn't match the real robot and is difficult to adjust in DiffTaichi. Thus, we excluded configurations prone to tipping from the training cases.

Moreover, the real robots are actuated using SMAs, which require a long cooling time to return to their original shape. However, in the simulation using DiffTaichi, the actuators return to their original shape instantaneously. To align the simulation with the real robot's behavior, we adjusted the control sequence in the simulation by removing the long waiting time necessary for the real robot. For the real robot, we maintained a consistent cooling period after each control action. This adjustment was applied uniformly across all configurations in the simulation.

V. EXPERIMENTS

We evaluate the performance of our proposed model using the five quantitative metrics outlined in Section III-B, complemented by qualitative analyses of representative

IF	PS	OPT	GEN	SR
Uni-Directional Locomotion				
70.00%	13.78	5.45	81.00%	98.00%
Back-and-Forth Locomotion				
61.00%	11.99	2.19	74.00%	89.00%
Downstairs Locomotion				
61.00%	16.73	4.11	71.00%	97.00%

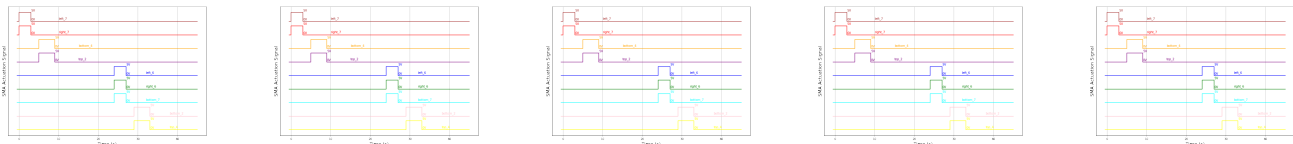TABLE I. Evaluation results of our model in three locomotion tasks.

generations. We use five prompts for each task objective and generate 10 robot designs per prompt for quantitative evaluation. The model employs a GPT-NEO [29] backbone and is trained using synthetic data across all three task objectives introduced in Section III-A.1. We aggregate the scores for each task objective and present them in Table I.

In response to all prompts, our proposed model consistently generates legal robot designs, ensuring that all modules are correctly interconnected without any "floating" or detached ones indicated by the relatively high success rate (SR) (over 89%) score. The instruction following (IF) score of over 60% shows the majority of robots designed by our model satisfy the structural and task-oriented requirements in the prompts.

The task optimality (OPT) and promise score (PS) metrics further show that the designed robots can complete the tasks within reasonable timeframes and have the capacity to travel longer distances than required within simulation time(on average 14.16 block lengths). For the generalizability (GEN) metric, which evaluates the creativity of robot-designing models, our model scores between 71% and 81%. This range indicates that the model generates robots distinct from the training data most of the time. Such results suggest our model's ability to infer the geometric and structural characteristics of superior robots rather than merely memorizing good robot designs in the training data.

To verify that the geometric and structural information of robots is captured and utilized by our model when producing robot designs, we use UMAP [30] to visualize the encoding of robot designs in the latent encoding space of our model. Our guiding hypothesis is that the similarity in the encodings of two robots indicates the model's perceived similarity between them. To preclude information leakage stemming from the binary choices training objective, we randomly select 1,000 robots not included in our model's training dataset and generate UMAP visualizations for these sampled robots. The UMAP visualization is displayed in Figure 6, with one representative robot design shown on the side of each cluster to reveal the clustering evidence.

The visualization indicates that our model effectively encodes the geometric characteristics of robot designs, e.g., the model distinctly clusters robots with varying aspect ratios within its latent space, suggesting the potential of our model to generate robot designs within complicated environments such as crawling through a small hole. Additionally, our model exhibits understandings of critical design elements such as the presence of legs and the optimal positioning of the robots' centers of mass. For example, designs with no

| (a) 5-module 2-leg | (b) 5-module 'L'-shape | (c) 4-module 1-chain | (d) 7-module 2-leg | (e) 8-module 3-leg |

Fig. 5. Control sequences for various robot configurations: (a) 5-module 2-leg, (b) 5-module 'L'-shape, (c) 4-module 1-chain, (d) 7-module 2-leg, and (e) 8-module 3-leg.
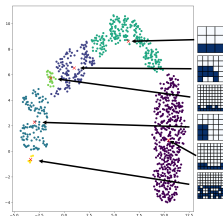

Fig. 6. Umap visualization of diverse robot designs encoded by our proposed model. Representative robot designs from the center of each cluster are also displayed.

leg, two legs, and even more legs are separated into distant clusters. These factors contribute to our model's designs in diverse environments, e.g., placing more weight at the rear end of the robot design to maintain balance while descending stairs and generating robots with more legs to facilitate faster movement on flat planes.

We further conduct manual validations to confirm our model's consideration of the task objective and the operating environment in its robot designs, which help it produce highly adaptable robots well-aligned with manual designs. Specifically, for unidirectional locomotion tasks, our model typically generates robots with right-side arms that can adjust the center of mass to accelerate movement (Figure 7a). In tasks requiring back-and-forth locomotion, the designs are more symmetrical, enabling efficient movement in both directions (Figure 7b). For the stair-descending task, the center of mass is strategically placed at the rear part of the robots, enhancing balance during the movement (Figure 7c).

VI. DISCUSSION

Section V has proven the overall strength of our proposed approach, and this section delves deeper into the contributing factors of our model's superior performance via a series of ablation studies.

A. Training Objective Ablation Experiments

As shown in Table II, ablating the CLM objective (**no-CLM**) leads to higher failure rates according to the IF and SR metrics and a comparable preference over task optimality, reflected in the PS and OPT scores. This suggests that the ablated model focuses less on representing legal and walk-able robot structures but instead prioritizes characteristics beneficial for accelerating the robots' movement. Conversely, the **no-Compare** model generally succeeds in designing robots as instructed in the prompts, though it occasionally overlooks the optimality of the designs.

This trend is supported by our qualitative assessments illustrated in Figure 7d, where the **no-Compare** model tends to produce more stable and symmetric designs that are, however, challenging to mobilize. On the other hand, the **no-CLM** model's creations often demonstrate more effective locomotion despite the high failure rate (Figure 7e). As such, we claim that incorporating both training objectives is crucial to ensure that the model adheres to instructions while also taking care of design quality,

B. Task Ablation Experiments

We further investigate the importance of task diversity to the generalizability of our model. Specifically, models trained on single tasks (**uni**, **back**, and **downstairs**) often perform well on their specific training objectives while suffer on tasks outside their training data. Introducing a second training objective (**uni-back**, **uni-downstairs**, and **back-downstairs** models) effectively narrows these performance gaps. These observations suggest that incorporating additional task objectives could enhance our models' capacity for solving other untested tasks, helping broaden the scope of robot designs using our model and leading to innovative and effective solutions difficult for experts to design.

Our manual examinations also reveal that the two-task models display better adaptability to the third task which they are not familiar with. For instance, the **uni-downstairs** model is prone to generating symmetric configuration to facilitate forward and backward walking (Figure 7i) and the **uni-back** model tends to increase weight distribution towards the back for enhanced stability when moving downstairs (Figure 7j). These observations are consistent with our quantitative findings, suggesting that training the models with multiple objectives fosters greater generalizability and encourages the models to generate robust and versatile robot designs for complex tasks.

C. Impacts of Training Data Sizes

We conduct repeated training and evaluations of our model using three different subsets of the training data, ranging from 40% to 80% in increments of 20%. The models corresponding to these data subsets are denoted as **0.4T**, **0.6T**, and **0.8T**. Our observations indicate that using less training data results in greater variability in the model's performance across different environments and task objectives. For instance, the **0.4T** model outperforms the **0.6T** and **0.8T** models in the stair descending task but significantly underperforms both models in the back-and-forth locomotion

	Uni-Directional Locomotion					Back-and-Forth Locomotion					Downstairs Locomotion				
	IF	PS	OPT	GEN	SR	IF	PS	OPT	GEN	SR	IF	PS	OPT	GEN	SR
our model	**70.00%**	13.78	5.45	81.00%	**98.00%**	**61.00%**	11.99	2.19	74.00%	89.00%	**61.00%**	16.73	**4.11**	71.00%	**97.00%**
no-CLM	41.00%	13.46	5.31	84.00%	92.00%	33.00%	9.42	2.16	80.00%	88.00%	36.00%	15.73	6.03	77.00%	85.00%
no-Compare	65.00%	13.83	5.24	83.00%	96.00%	53.75%	8.75	2.34	73.75%	**93.75%**	48.00%	16.15	5.46	81.00%	93.00%
uni	69.00%	**14.60**	5.24	**93.00%**	**98.00%**	53.00%	10.76	2.20	**85.00%**	89.00%	48.00%	12.14	5.20	**87.00%**	76.00%
back	61.00%	13.41	4.82	81.00%	91.00%	45.00%	10.21	2.07	**85.00%**	91.00%	48.00%	15.48	5.62	86.00%	**97.00%**
downstairs	57.00%	11.82	4.64	78.00%	90.00%	54.00%	10.48	**1.96**	77.00%	89.00%	44.00%	13.86	5.18	70.00%	84.00%
uni+back	69.00%	4.51	5.07	**90.00%**	95.00%	55.00%	1.99	2.54	**95.00%**	**95.00%**	48.75%	13.12	**5.17**	**90.00%**	92.50%
uni+downstairs	**73.00%**	13.73	**4.39**	89.00%	97.00%	**57.50%**	4.54	2.16	67.50%	90.00%	45.00%	8.86	**4.40**	81.67%	**100.00%**
back+downstairs	68.00%	9.73	**4.47**	70.00%	88.75%	53.00%	1.83	2.41	**90.00%**	**95.00%**	42.00%	15.74	5.47	86.00%	**97.00%**
GPT-2	**70.00%**	**14.41**	5.47	75.00%	**98.00%**	50.00%	10.17	**1.90**	77.00%	90.00%	**53.00%**	17.24	6.55	76.00%	96.00%
BART	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
0.8T	68.00%	13.68	5.00	78.00%	94.00%	**62.00%**	10.93	2.23	73.00%	88.00%	40.00%	14.17	6.28	75.00%	90.00%
0.6T	**70.00%**	14.02	4.76	75.00%	96.00%	56.00%	**12.08**	2.02	71.00%	93.00%	48.00%	15.81	6.40	72.00%	**97.00%**
0.4T	64.00%	**14.25**	**4.62**	**90.00%**	97.00%	49.00%	10.59	**1.75**	83.00%	89.00%	**58.00%**	17.89	5.46	**91.00%**	**97.00%**

TABLE II. Evaluation results in main and ablation experiments. Top 3 scores of each metric are bolded.

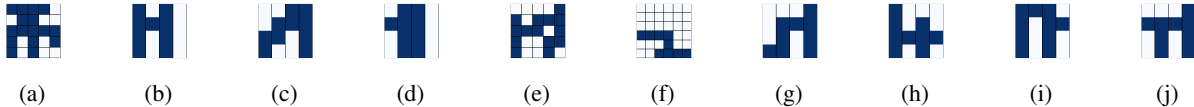| (a) | (b) | (c) | (d) | (e) | (f) | (g) | (h) | (i) | (j) |

Fig. 7. Examples of different models and tasks: (a)-(c) Our model tasks for unidirectional, back-and-forth, and descending locomotion; (d)-(e) No-Compare and No-CLM models; (f)-(h) Downsampled training data examples; (i)-(j) Uni-downstairs and Uni-back tasks.

task. Additionally, manual analysis of the robot designs generated by these models reveals that robot designs produced by the **0.4T** model are often unstable or impractical for real-world applications, as illustrated in Fig. 7f. In contrast, the additional training data used in the **0.6T** and **0.8T** models contribute to training models with more stable performance across tasks and settings. It is noteworthy from these results that the performance gain of the **0.8T** model over the **0.6T** model is only marginal, suggesting the data efficiency of our proposed approach.

D. Impacts of Base Model Selection

Selecting robust backbone models that can comprehend our instructions and generate suitable robot designs is essential in robotics research. Our preliminary experiments utilized two smaller models: **BART** [31] and **GPT-2** [32], where **BART** is a masked language model and **GPT-2** is a CLM. We note that **BART** struggled across all trials, never producing valid natural language expressions either before or after training. Conversely, the **GPT-2** model performed well in generating robots for the simpler unidirectional locomotion task but fell short on the other two objectives, highlighting the limitations of smaller models in mastering the complex robot design generation task. Consequently, we shifted to using a larger LLM, i.e., **GPT-NEO**, which demonstrated a robust understanding of our prompts and effectively generated accurate, high-quality robot designs. While the exploration of larger LLMs appears promising, in future work it is also important to weigh the substantial demands of training time, evaluation overhead, and data consumption against the performance improvements.

VII. Conclusion

We introduced a new approach combining pre-trained language models (LLMs) and differentiable physics simulation tools to automate the design of soft modular robots tailored to specific environments and task objectives. To evaluate the approach, we propose five metrics that assess the quality and generalizability of these designs in different physical contexts. We demonstrate the efficacy of our approach by creating appropriate robots following natural language instructions to accommodate different design tasks.

VIII. Limitations and Future work

One limitation of our current system is the slow cooling time of the Shape Memory Alloy (SMA) coils, which restricts the locomotion speed of the robot. Although SMAs were chosen for their favorable force-to-mass ratio and high work density—making them effective for quickly and efficiently demonstrating our concept—this slow cooling time is a significant drawback for practical applications. To address this limitation, we are exploring the replacement of SMAs with alternative actuators, such as motors, which could increase actuation speed and allow for untethered operation. This change would enhance the adaptability of the modules to more complex environments, making the system more suitable for practical applications.

We used computer vision with color-coded markers to track the (x, y) positions of all robot vertices. However, discrepancies exist between the real and simulated robots: real-world connections have four nodes on the connecting line, while the simulation, limited by DiffTaichi, has only two. Adjusting the simulation would require major changes to DiffTaichi's core. Perfect alignment remains challenging, especially as simulations show more bounce than real tests, which cannot be easily corrected through parameter tuning.

To address these challenges, we are developing a specialized physical engine tailored to our robots' specific physical properties to enhance simulation fidelity. While the primary objective of this paper is to validate the use of learning-based methods for designing soft modular robots, narrowing the gap between simulated and real-world performance remains a key focus for our future work.

As this is the first work to use natural language for designing soft modular robots, there is no baseline for direct comparison. The closest research involves rigid modular robots, which have smaller search spaces due to fixed block functionality and limited connection options. Thus, we com-

pare our full model to ablated versions—where specific data or objectives are removed—to demonstrate the importance of each component and highlight the model's capabilities.

In future work, we plan to extend our approach to the design of other types of robots, where comparisons with state-of-the-art models in those domains will allow us to further demonstrate the generalizability of our proposed method. However, such comparisons are beyond the scope of this paper, which primarily serves as a proof of concept. Our focus here is to validate the feasibility of using natural language for representing soft modular robot configurations and to explore the potential of pre-trained language models (LLMs) in the robot design process.

REFERENCES

[1] Chao Zhang, Pingan Zhu, Yangqiao Lin, Zhongdong Jiao, and Jun Zou. "Modular Soft Robotics: Modular Units, Connection Mechanisms, and Applications". In: *Advanced Intelligent Systems* 2 (2020).

[2] Luyang Zhao, Yitao Jiang, Chun-Yi She, Muhao Chen, and Devin Balkcom. *SoftSnap: Rapid Prototyping of Untethered Soft Robots Using Snap-Together Modules*. 2024. arXiv: 2410.19169 [cs.RO].

[3] Luyang Zhao et al. "Soft Lattice Modules That Behave Independently and Collectively". In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 5942–5949.

[4] Julian Whitman, Raunaq M. Bhirangi, Matthew J. Travers, and Howie Choset. "Modular Robot Design Synthesis with Deep Reinforcement Learning". In: *AAAI Conference on Artificial Intelligence*. 2020.

[5] Allan Zhao et al. "RoboGrammar: graph grammar for terrain-optimized robot design". In: *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: 10.1145/3414685.3417831.

[6] Jiaheng Hu, Julian Whitman, Matthew J. Travers, and Howie Choset. "Modular Robot Design Optimization with Generative Adversarial Networks". In: *2022 International Conference on Robotics and Automation (ICRA)* (2022), pp. 4282–4288.

[7] Tzuf Paz-Argaman, John Palowitch, Sayali Kulkarni, Reut Tsarfaty, and Jason Baldridge. "Into the Unknown: Generating Geospatial Descriptions for New Environments". In: *Findings of the Association for Computational Linguistics ACL 2024*. 2024, pp. 2259–2273.

[8] Zixia Jia, Mengmeng Wang, Baichen Tong, Song-chun Zhu, and Zilong Zheng. "LangSuit· E: Planning, Controlling and Interacting with Large Language Models in Embodied Text Environments". In: *Findings of the Association for Computational Linguistics ACL 2024*. 2024, pp. 14778–14814.

[9] John W. Romanishin, Kyle Gilpin, and Daniela Rus. "M-blocks: Momentum-driven, magnetic modular robots". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 4288–4295. DOI: 10.1109/IROS.2013.6696971.

[10] S. Hauser, M. Mutlu, P.-A. Léziart, H. Khodr, A. Bernardino, and A.J. Ijspeert. "Roombots extended: Challenges in the next generation of self-reconfigurable modular robots and their application in adaptive and assistive furniture". In: *Robotics and Autonomous Systems* 127 (2020), p. 103467. ISSN: 0921-8890.

[11] Mark Yim et al. "Modular self-reconfigurable robot systems [grand challenges of robotics]". In: *IEEE Robotics & Automation Magazine* 14.1 (2007), pp. 43–52.

[12] Davide Zappetti, Stefano Mintchev, Jun Shintake, and Dario Floreano. "Bio-inspired tensegrity soft modular robots". In: *Conference on Biomimetic and Biohybrid Systems*. Springer. 2017, pp. 497–508.

[13] Nathan S. Usevitch, Zachary M. Hammond, Mac Schwager, Allison M. Okamura, Elliot W. Hawkes, and Sean Follmer. "An untethered isoperimetric soft robot". In: *Science Robotics* 5.40 (2020), eaaz0492.

[14] Stefano Mintchev et al. "An underwater reconfigurable robot with bioinspired electric sense". In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 1149–1154.

[15] Shunichi Kurumaya et al. "A modular soft robotic wrist for underwater manipulation". In: *Soft robotics* 5.4 (2018), pp. 399–409.

[16] Qiji Ze et al. "Soft robotic origami crawler". In: *Science advances* 8.13 (2022), eabm7834.

[17] Shuguang Li et al. "Scaling up soft robotics: A meter-scale, modular, and reconfigurable soft robotic system". In: *Soft Robotics* 9.2 (2022), pp. 324–336.

[18] Matthew A Robertson and Jamie Paik. "New soft robots really suck: Vacuum-powered systems empower diverse capabilities". In: *Science Robotics* 2.9 (2017), eaan6357.

[19] Cagdas D. Onal and Daniela Rus. "A modular approach to soft robots". In: *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 2012, pp. 1038–1045.

[20] Jun-Young Lee, Woong-Bae Kim, Woo-Young Choi, and Kyu-Jin Cho. "Soft Robotic Blocks: Introducing SoBL, a Fast-Build Modularized Design Block". In: *IEEE Robotics Automation Magazine* 23.3 (2016), pp. 30–41.

[21] Steven Ceron, Marta An Kimmel, Alexandra Nilles, and Kirstin Petersen. "Soft Robotic Oscillators With Strain-Based Coordination". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7557–7563.

[22] Andrea Vergara, Yi-Sheng Lau, Ricardo-Franco Mendoza-Garcia, and Juan Cristóbal Zagal. "Soft modular robotic cubes: Toward replicating morphogenetic movements of the embryo". en. In: *PLoS One* 12.1 (Jan. 2017), e0169179.

[23] Jun Zou, Yangqiao Lin, Chen Ji, and Huayong Yang. "A reconfigurable omnidirectional soft robot based on caterpillar locomotion". In: *Soft Robot.* 5.2 (Apr. 2018), pp. 164–174.

[24] Luyang Zhao et al. "StarBlocks: Soft Actuated Self-Connecting Blocks for Building Deformable Lattice Structures". In: *IEEE Robotics and Automation Letters* 8.8 (2023), pp. 4521–4528. DOI: 10.1109/LRA.2023.3284361.

[25] Chiwon Lee et al. "Soft robot review". In: *International Journal of Control, Automation and Systems* (2017).

[26] Yuanming Hu et al. "DiffTaichi: Differentiable Programming for Physical Simulation". In: *International Conference on Learning Representations*. 2019.

[27] Alec Radford. "Improving language understanding by generative pre-training". In: (2018).

[28] Henry Segerman and Rosa Zwier. "Magnetic Sphere Constructions". In: 2017.

[29] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. Version 1.0. Mar. 2021. DOI: 10.5281/zenodo.5297715. URL: https://doi.org/10.5281/zenodo.5297715.

[30] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018).

[31] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703.

[32] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners". In: (2019).