Generative-AI-Driven Jumping Robot Design Using Diffusion Models

Byungchul Kim*, Tsun-Hsuan Wang*, and Daniela Rus

***Abstract*— Recent advances in foundation models are significantly expanding the capabilities of AI models. As part of this progress, this paper introduces a robot design framework that uses a diffusion model approach for generating 3D mesh structures. Specifically, we focus on generating directly fabricable robot structures that require no post-processing guided by human-imposed design constraints. Our approach can find the optimal design of the robot by optimizing or composing embedding vectors of the model. The efficacy of the framework is validated through an application to design, fabricate, and evaluate a jumping robot. Our solution is an optimized jumping robot with a 41% increase in jump height compared to the state-of-the-art design. Additionally, when the robot is augmented with an optimized foot, it can land reliably with a success ratio of 88% in contrast to the 4% success ratio of the base robot.**

***Index Terms*— Computational Design, Generative Artificial Intelligence, Design Optimization, Jumping Robot**

I. Introduction

Robotics is rapidly advancing due to innovations in both hardware and software. Improved actuators [1]–[3] and sensors [4] have expanded robot capabilities, while AI technology—bolstered by increased computational power and large datasets—has accelerated progress in robot control for complex, high-level tasks [5]–[7].

Robot hardware design, unlike software development, often relies on traditional methods due to the complex, multi-domain design space it encompasses, including dynamics, kinematics, material science, and fabrication techniques. Past research has advanced this field through bio-inspired robots [8], [9], which emulate natural motion principles, and soft robots [10], [11], which use flexible materials to introduce embodied intelligence. Additionally, optimal design studies focus on selecting and mathematically optimizing design parameters based on their relationship to robot performance [12]. While these methods have led to innovative breakthroughs, they require substantial human effort to fully understand the robot's operating principles and to develop appropriate designs, which slows down the hardware design (and fabrication) iteration process compared to the rapid pace of software development.

Recently, sampling-based design methods have been introduced to accelerate design iteration. For example, new robot designs can be generated by combining design features from a set of existing robot designs [13] or foundational models [14]. The incorporation of simulation tools that evaluate the performance of the robot enables these sampling methods to rapidly explore optimal designs. Due to their fast iteration and generalizability, these methods have shown significant potential in multi-domain optimization. For example, it is possible to optimize the control method simultaneously with design, often referred to as co-design or designing specific robot components such as soft actuators to improve the robot's performance [15]. However, sampling-based methods face challenges in considering fabrication and assembly constraints, as well as in integrating traditional manufacturing techniques and mechanical components, particularly in areas where large, relevant datasets are often lacking.

In this paper, we present a diffusion model-based robot design framework that bridges the gap between traditional and sampling-based design methods. This framework generates optimal robot structures that are directly fabricable without the need for post-processing by building them on top of a human-designed base robot, as shown in Fig. 1. To validate the effectiveness of the approach, we apply it to jumping robots—a domain where the dynamic nature of motion highlights the critical importance of design optimization[16], [17]. The paper contributions are:

- a method for seamlessly transforming mesh structures sampled from generative models into human-designed robot structures, eliminating the need for post-processing.
- an optimization method that enhances the generative model's ability to produce robot structures, improving performances across both single-objective and multi-objective functions.
- a composition method, inspired by [18], for generating new designs that enhance overall robot performance without additional optimization by composing pre-optimized embedding vectors, each trained for different objectives.
- application of the 3 methods to the design and fabrication of a jumping robot optimized to achieve maximal jumping height and a stable landing.

II. Related Work

Jumping Robots. Jumping is an intriguing form of locomotion for robots, as it enables them to overcome obstacles or reach elevated areas that may be challenging for other modes of movement [19]. Achieving an effective jump, which involves rapid body acceleration in a short time, has

This work was supported by the NSF EFRI Program (Grant No. 1830901), the SMART M3 program, and the GIST-CSAIL Collaboration program.

B. Kim, T.-H. Wang and D. Rus are with the Distributed Robotics Laboratory, Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139, USA. * indicates equal contributions. (email: bckim@csail.mit.edu, tsunw@mit.edu, and rus@csail.mit.edu)

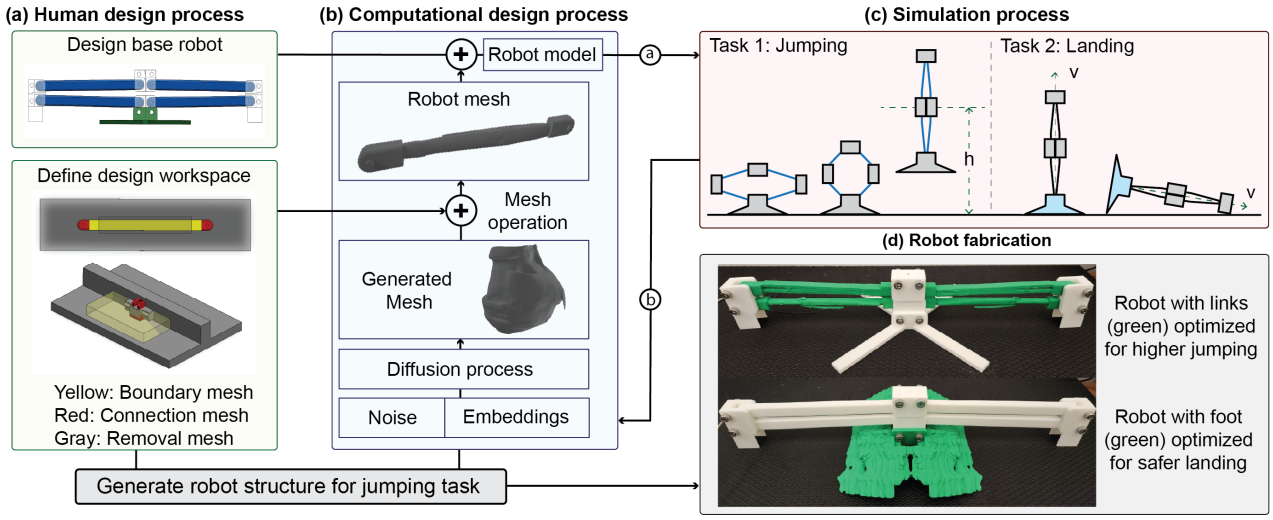

Fig. 1. **An overview of the proposed framework. (a)** The human first designs the base jumping robot with 3D CAD software (Solidworks, Dassault Systems) and specifies the design workspace where the computational design will be conducted or not. **(b)** Given human-guided designs as an input, the pre-trained diffusion model generates 3D structures that will be aligned to the robot structure. **(c)** The simulator (MuJoCo) is used measures the performance of the robot generated from the computational design block (arrow a). In this research, jumping and landing performance are measured. The performance is fed back to the computational design block (arrow b) to find proper embedding vectors that generate a high-performance robot design. **(d)** The real-world robot is fabricated using the optimal embedding sets. The robot design from the previous blocks doesn't need any post-processing as it is generated on top of the base robot designed by the human.

driven researchers to explore efficient robotic mechanisms and innovative actuation methods [20]–[24]. Nature has also served as inspiration, with robots designed based on animals such as fleas [25], water striders [17], and locusts [26], which exhibit evolved strategies for jumping.

Generative Approaches for Robot Design. The tremendous success of generative AI in other domains like natural language [27], imagery [28], and 3D assets [29], have inspired other related fields to adopt the generative approach. Specifically, in robot design, with the help of physics simulation [30]–[32], methods are developed for grippers [33], modular robots [34], locomotion [35], or general co-design [14] to leverage generative models as a source to propose diverse and functional solutions.

III. Robot Design and Optimization

A. Robot Design as a Computational Problem

We can write robot design problem as a computational problem $\max_{\psi\in\Psi} f(\psi)$, where Ψ is the design space and $f(\cdot)$ measures the performance of the robot. The main challenge is the extremely large search space in Ψ, rendering the optimization intractable. We propose to alternatively tackle an approximation of the original problem,

$$\max_{\psi\sim p^*(\psi)} f(\psi), \quad \text{where } p^* = \max_{p\in\mathscr{P}} \mathbb{E}_p[f(\psi)] \tag{1}$$

where $\mathscr{P}$ is a set that covers all possible distributions in the search space Ψ. The problem is broken down into finding a "good" distribution $p^*(\psi)$ which allows for the sampling of well-performing robots and performing maximization over it. This formulation enables us to leverage the recent advancement of generative AI [36], [37], where we can sample from the pre-trained generative model $\psi\sim p(\psi)=\mathbb{E}_{p_\theta(\mathbf{x})}[p(\psi|\mathbf{x})]$ with $p_\theta(x)$ as some parts of the design such as the link shape of an articulated robot. Second, we can similarly constrain the search space with human design $\mathscr{P}_{\text{human}}\subseteq\mathscr{P}$ as in the original formulation $\Psi_{\text{human}}\subseteq\Psi$, e.g., only considering the space with certain linkage topology designed by human expert. Third, with certain types of generative models, we can compose distributions $p_\theta(\mathbf{x}|\mathbf{c}_1,\mathbf{c}_2,\dots)\propto p_\theta(x,\mathbf{c}_1,\mathbf{c}_2,\dots)=p(\mathbf{x})\prod_i p(\mathbf{c}_i|\mathbf{x})$ [38]; for example, $p(\mathbf{x}|\mathbf{c}_1)$ may be a link shape distribution for jumping higher and $p(\mathbf{x}|\mathbf{c}_2)$ may that for landing more stably.

B. Diffusion-based Generative Models for Shape Design

Diffusion generative models [39]–[41] are latent variable models as $p(\mathbf{x})=p(\mathbf{x}_0)=\int p(\mathbf{x}_T)\prod_{t=1}^T p(\mathbf{x}_{t-1}|\mathbf{x}_t)d\mathbf{x}_{1:T}$. The transition kernel in the forward process is defined as a Gaussian diffusion process $q(\mathbf{x}_t|\mathbf{x}_{t-1})=\mathcal{N}(\mathbf{x}_t;\sqrt{1-\beta_t}x_{t-1},\beta_t\mathbf{I})$ for some $0<\beta_t<1$, starting at $p(\mathbf{x}_T)=\mathcal{N}(\mathbf{x}_T;\mathbf{0},\mathbf{I})$. This allows an analytical form that corrupt the clean data $\mathbf{x}_0$ with noise $q(\mathbf{x}_t|\mathbf{x}_0)=\mathcal{N}(\mathbf{x}_t;\sqrt{\bar\alpha_t}\mathbf{x}_0,(1-\bar\alpha_t)\mathbf{I})$, where $\bar\alpha_t=\prod_{s=1}^t\alpha_s$ and $\alpha_t=1-\beta_t$. The training is to learn the reverse diffusion process via a so-called denoiser ε_θ with the loss,

$$\min_\theta \mathbb{E}_{t\in[1,T],p(\mathbf{x}_0),\mathcal{N}(\varepsilon;\mathbf{0},\mathbf{I})}[||\varepsilon-\varepsilon_\theta(\mathbf{x}_t,t)||^2]. \tag{2}$$

During inference time, we first sample from random noise $\mathbf{x}_T\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ and gradually apply the denoiser

$$\mathbf{x}_{t-1}\sim\mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\varepsilon_\theta(\mathbf{x}_t,t)\right),\frac{1-\bar\alpha_{t-1}}{1-\bar\alpha_t}\beta_t\mathbf{I}\right) \tag{3}$$

We are using the Shap-E model [37], where diffusion operates on the parameterized implicit function of 3D shape – we denote the generated output directly as 3D shapes for simplicity. In our case, the sampling of the robot designs $\psi\sim p(\psi)$ then becomes, based on the linkage topology from human conceptual design $p(\psi|\cdot)$, a sample of the shapes of different links modeled by the diffusion model $\mathbf{x}\sim p_\theta(\mathbf{x})$.

The sampled shapes undergo automated processing to construct mesh structures for seamless integration into a human-designed robot. This transformation involves a series of mesh operations using three reference meshes—referred to in this paper as the boundary, removal, and connection meshes. First, the generated mesh is transformed to ensure its bounding box fits the *boundary mesh*. Next, it is refined using the *removal mesh* to avoid collisions with other components and ensure smooth fabrication and assembly. Finally, the refined mesh is integrated into the robot's structure by combining it with the *connection mesh*.

C. Embedding Optimization

Generative models can normally take embeddings $\mathbf{c}$ as inputs for conditional generation $p(\mathbf{x}|\mathbf{c})$, e.g., using text embedding to do text-to-3D generation. Prior work [14] proposed to train the embedding for improved physical utilities using physics-based simulation. The high-level idea is to (i) actively create new data from the model and store it in a buffer, (ii) leverage physics-based simulations to measure performance, and (iii) optimize embeddings conditioned on the diffusion model under a biased data distribution to enhance robot performance. We adapt this technique in our pipeline and extend it to ensure the feasibility of fabrication,

$$\min_{\mathbf{c}} \mathbb{E}_{t\sim[1,T],p_\theta(\mathbf{x}_0|\mathbf{c}),\mathcal{N}(\varepsilon;\mathbf{0},\mathbf{I})}[||\varepsilon-\varepsilon_\theta(\mathbf{x}_t,t,\mathbf{c})||^2]. \quad (4)$$

The major difference is how $p_\theta(\mathbf{x}_0|\mathbf{c})$ is constructed – the online training data generation, relevant to steps (i) and (ii) mentioned above. First, we filter out all generated data that violates the fabrication constraints including penetrating other links or the environment in the initial condition, not being watertight, and being too thin. Next, we evaluate the performance of the design with a physics engine. Finally, if we optimize for multiple objectives, we find the Pareto front using linear regression, filtering out samples with performance below the lower bound of a certain objective, and find the farthest point set with respect to the Pareto front. The evaluation of robot performance and the feasibility constraints are implemented using MuJoCo [30] with the linkage topology design represented as URDFs.

D. Composition of Embedding Vectors

With the formulation in Sec. III-A, we can compose robot designs by composing the distributions. For example, we can compose a distribution for jumping higher and another for landing more stably into a distribution that generates a design for simultaneously jumping high and landing stably. With diffusion-based generative model, we can achieve compositional design by (i) performing embedding optimization on different objectives, giving $\mathbf{c}_1,\mathbf{c}_2,\ldots$ and (ii) sample the compositional distribution via the diffusion generative process as in [38] via modifying the denoising/score term ε_θ in Eq. 3 to $\hat{\varepsilon}(\mathbf{x}_t)=\varepsilon_\theta(\mathbf{x}_t,t)+\sum_i \varepsilon_\theta(\mathbf{x}_t,t,\mathbf{c}_i)$. More advanced methods to sample from a compositional distribution such as [42], [43] are left to future exploration.

IV. Automatic Design of Optimized Jumping robot

A. Base robot design

As our framework requires base robot design as an input, we have designed the base robot using Solidworks (Dassault Systems) inspired by previous jumping robots [20]–[22]. It has a rhombus-shaped structure (Fig.2a) that can rapidly move its center of mass (COM) vertically. Before jumping, the upper and lower links are parallel to the ground. The external motor pulls the tendon, which passes TR1 and TR2, making them perpendicular to the ground. This action raises the robot, shifting its COM vertically and initiating the jump as illustrated in Fig.1(c). The foot is intuitively designed with a cross shape to provide sufficient support for the robot's body. Since our research focuses on discovering improved link shapes using generative AI, we located actuators and other circuits outside the robot to minimize their impact on jumping performance and facilitate rapid design iteration.

The MuJoCo model for the simulation is also built using the base robot. Due to the limitation of the URDF which doesn't support parallel linkages, we modified the robot model by dividing the robot foot to become two feet while fabricating the real-world robot with a single foot. To reduce the sim-to-real gap as much as possible, we applied equality conditions between two feet geometries.

B. From components to the jumping robot

We augmented the upper and lower link designs to enhance jumping height (single objective) and the foot designs to improve landing stability while maintaining jumping performance (multi-objective). Three reference meshes, that are used to transform generated raw meshes into fabricable robot structures, are defined for links and foot, as shown in Fig. 2(b) and (c), respectively. For the links, the boundary mesh (yellow in the figure) is designed to be centered on the base link with a cuboid shape, while the connection mesh (red in the figure) is designed to have the same shape with both ends of the base link. To ensure reliable integration, the boundary and connection meshes are slightly overlapped, preventing the final link from becoming too thin. A removal mesh was designed to eliminate any structures that could interfere with the robot's motion by causing collisions with other parts.

The boundary and connection meshes for the foot were similarly defined to provide design guidance. The distinct difference was in the removal mesh as shown in the enlarged view of Fig. 2(c)–this mesh was designed with space to accommodate bolts and nuts, ensuring the foot can be assembled with other links without requiring post-processing.

We conducted mesh operation for the upper and lower links and one foot of the robot, while the opposite-side meshes were mirrored to ensure a symmetric robot design.

C. Embedding Optimization for Link Design

Optimization for the embedding vectors of the links focused on maximizing the robot's maximum jumping height (MJH). Since an optimal design for the upper link may not be equally effective for the lower link, we sampled distinct

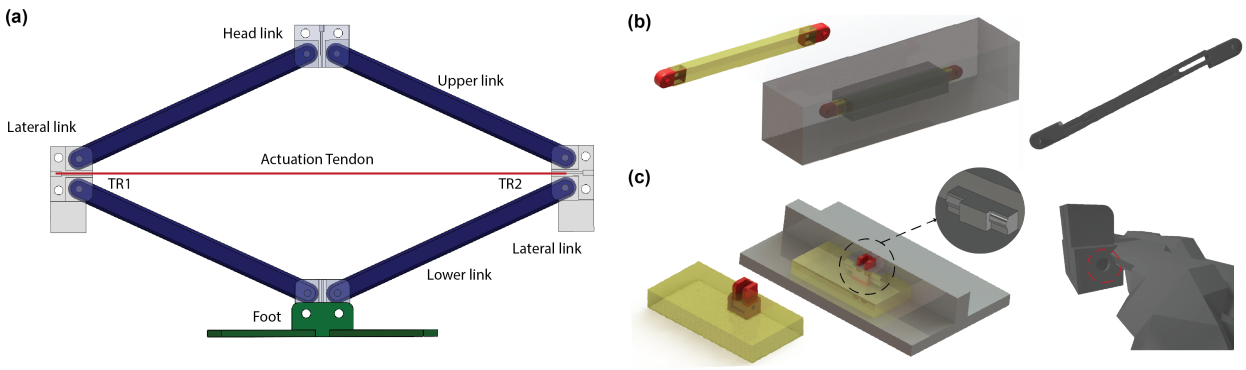

Fig. 2. **An overview of the base robot and pre-defined design workspace for the mesh transformation. (a)** shows the overall design of the base robot, which serves as the input for the proposed framework. In this paper, designs for the upper and lower links, as well as the foot, are generated based on this base design. **(b)** and **(c)** illustrate the design workspace for the robot's link and foot, respectively. These workspaces, defined by the human designer considering fabrication and assembly, highlight areas for boundary (yellow), connection (red), and removal (gray) within the base design, which serve as constraints for the AI-driven design process. The right side of the figures shows the example of generated mesh from this mesh operation process.

designs from separate embedding vectors for each. The z-axis position of the approximate center of mass (COM) is calculated in the simulation to find the MJH of the robot. To balance computational resources, we set the number of samples per optimization run to 500 and the filter size to 12.

One potential issue is that the link shape could converge to an extremely thin design, as a lightweight structure may offer benefits in achieving greater jumping heights. However, such designs can be challenging to fabricate in real-world. To address this, we estimated the mesh thickness by calculating the average cross-sectional area of the mesh perpendicular to its first principal axis. If this area is less than 40% of that of the base link, we assigned their MJH to zero to filter them out. Similarly, meshes with defects, such as non-watertight structures, were also filtered by setting their MJH to zero.

D. Embedding Optimization for Foot Design

The robot's foot structure is optimized focusing on landing stability, as it plays a critical role during the landing phase. To isolate the impact of foot design, all other robot components remain consistent with the base model. Landing stability is validated using the z-component of the body direction vector after jumping, where a value of 1 and 0 each indicates the robot is standing upright or has fallen.

While the robot's symmetrical design may result in successful landings in simulation, real-world factors such as friction, ground slope, and fabrication imperfection may increase the risk of falling. To improve landing reliability under these disturbances, we simulated its landing in seven scenarios: one baseline condition with no external bias, three with mass biases applied to the upper, lateral, and lower links, and three with altered tendon routing, where tendon points were slightly shifted to induce intentional asymmetry. The final landing stability was calculated by averaging the body direction vectors across all seven conditions.

Recognizing that optimizing for landing stability could lead to designs that reduce MJH, we adopted a multi-objective optimization approach, balancing both landing stability and jumping performance, for the foot design: the foot design may converge toward a heavier structure, as increased weight helps stabilize landings. To address the trade-off between these metrics, we first performed linear regression on the data after excluding non-watertight mesh cases. We then filtered out designs with a body direction value lower than 0.85 because landing stability was prioritized here. From the refined dataset, we selected designs that deviated positively from the regression line for the optimization, to find the structure that offers consistent landing stability without compromising jump height.

V. Results

A. Embedding Optimization results for Link Design

The optimization results for the link design (Fig. 3) show how the MJHs change as the optimization number increases. First, we can find that the maximum value of MJH increases each time the optimization is repeated. This means that, through optimization, a better design that is not sampled without optimization can be generated after the optimization. We can also see that the sampling distribution (at the right side of each graph) has moved upwards. It shows that the optimization actually moves the average of the distribution, improving the sampling efficiency - i.e., the optimization creates a better chance to sample robot designs that can jump better. Another interesting result here is that the optimization reduces the number of zero jumping heights; it means that the optimization is reducing the likelihood of making fault meshes because a jumping height of zero means that there are defects in the generated meshes.

B. Embedding Optimization results for Foot Design

Fig. 4 shows how the MJH and stability change through the optimization of the foot design. The top 12 designs sampled from the non-optimized distribution can only jump about 0.4 meters, but the top results move to the right side after repeated optimizations–this can also be seen in the distribution graphs at the top of the figure. Since the best result is observed in the design sampled from the embedding that was optimized two times, we used this design when developing the real-world robot.

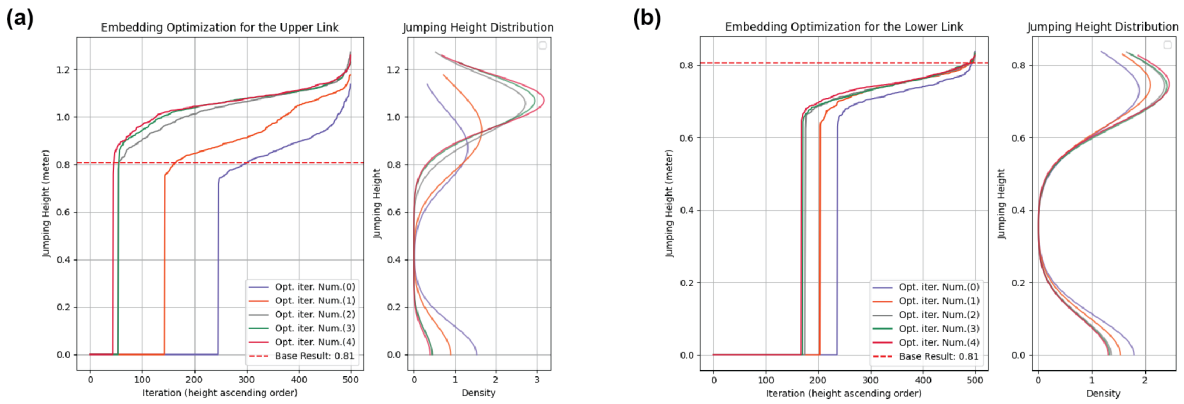

Fig. 3. **Embedding Optimization Results for Upper and Lower Links Design. (a)** and **(b)** present the results of embedding optimization for the upper and lower links, respectively, with a focus on maximizing jumping height.

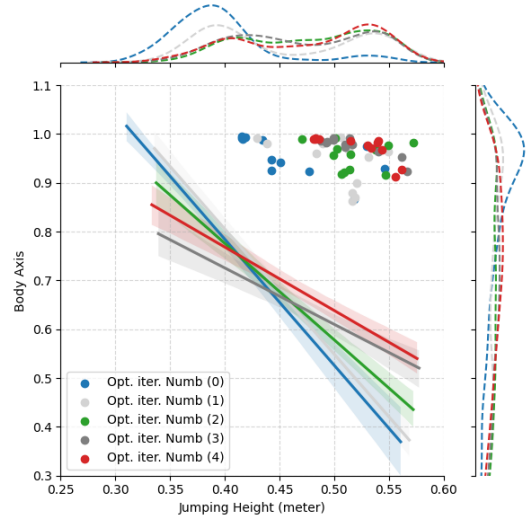

Fig. 4. **Embedding Optimization Results for Foot Design.** Optimization of embedding vectors that sample foot design for two objectives: jumping height (x-axis) and landing stability (y-axis).

C. Embedding Composition results for Foot Design

We evaluated how the composition of embedding vectors, each optimized with max jump height and landing stability, can be used in the proposed design pipeline. As shown in yellow and purple distributions in Fig. 5(a), each embedding vector generates designs biased to one objective. However, after the composition, the new embedding generates designs that can show good results in both objectives as shown in green data points and distribution without additional multi-objective optimization.

While the composition results showed slightly lower performance compared to previous multi-objective optimizations (red data points in the graph), embedding composition remains a promising approach with significant potential in robot design. One potential benefit is the ability to optimize embeddings for specific robot performance features individually, without the complexity of multi-objective filtering. These embeddings can then be composed based on the desired priorities, allowing us to generate new robot designs without the need for additional multi-objective optimization. For example, we can see the sampled distributions move from the stability-biased distribution (Fig. 5b) to the max-height-biased distribution (Fig. 5d) by changing the guidance levels of two embeddings without additional optimizations.

D. Real-world robot experiment

To validate the effectiveness of our framework, we fabricated robots using a Bambu Lab P1S 3D printer with polylactic acid (PLA Basic, Bambu Lab) filament. Since our research focuses on structural design, the motor (AK60-6 V1.1 24V Exoskeleton Module-KV80, T-Motor) was positioned externally to eliminate the influence of motor dynamics during testing. Tension was transmitted via a Bowden cable, and all hardware experiments were conducted using position control with maximum acceleration and velocity. In line with the simulations, the maximum jumping height was assessed by estimating the COM height based on the position of the two lateral links. A demonstration of both the base and AI-generated robots jumping is shown in Fig. 6.

The average MJH is obtained over 25 experimental trials for robots with different link designs (Table I) with the values in parentheses representing the standard deviation. Comparing real-world data with simulation results reveals a noticeable sim-to-real gap. Despite this discrepancy, the robot with the optimized link design shows significant improvement in jumping height. Interestingly, while the real-world results are approximately 55% of the simulation values, the trend between the two remains consistent. This suggests that our design pipeline can still be effective, provided the sim-to-real gap remains predictable.

The robot with optimal foot design landed safely in 22 out of 25 jumping trials, while that with the base foot landed safely only once in 25 trials. One possible reason for the three failed landings in the real-world tests could be the influence of the spring sheath attached to the lateral link for actuation. For more details, please refer to the supplementary video showing the successful and failed cases. As expected, the optimized foot reduced the jumping height due to its increased weight: the robot with the optimized foot jumped 0.36 meters, whereas the robot with both optimized links and foot jumped 0.42 meters.

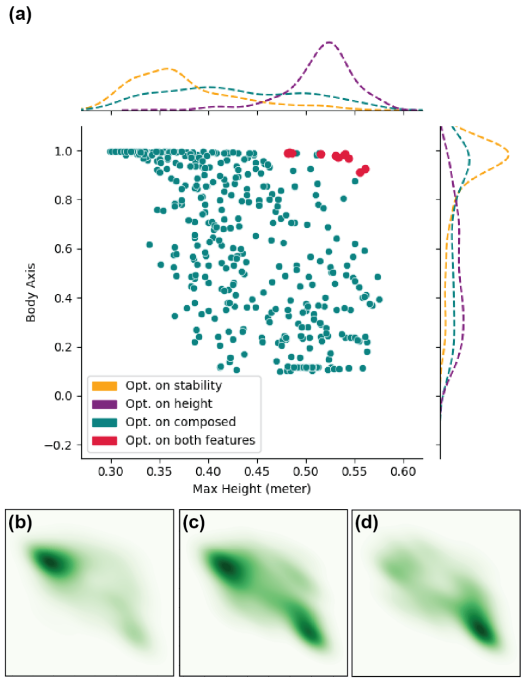

Fig. 5. **Embedding Composition Results for Foot Design. (a)** shows the performance distribution of designs sampled from various embeddings. The yellow and purple distributions represent the performance of designs sampled from embeddings optimized for a single objective. The green scatter plot and distribution represent the performance of designs sampled from a composition of two embeddings. Red scatter points indicate the performance of designs sampled from embeddings optimized for multiple objectives. **(b)** - **(d)** present the performance distributions of designs from stability-biased, balanced, and jumping-biased guidance, respectively. The x- and y-axes are omitted for clarity but are consistent with those in **(a)**.

TABLE I

SIMULATION AND EXPERIMENTAL RESULTS OF ROBOT JUMPING

	Base link	**Sampled link**	**Optimized link**
Sim.	0.74m	1.04m	1.19m
Real	0.39m (0.11)	0.46m (0.15)	0.55m (0.12)

E. Possible Reasons for the higher performance

Although our design pipeline does not explicitly explain why the generated designs perform better, several factors can be inferred from quantitative and qualitative observations of the simulation. To understand these improvements, it is essential to understand the dynamics of jumping locomotion [44]: (1) The input work applied by the motor before the jump affects the jumping height. (2) The efficiency of converting input work into kinetic energy (in a vertical direction) is key-—this energy must primarily translate into vertical, rather than horizontal, direction; and (3) At the takeoff moment, the combined vertical momentum of all body parts determines the robot's COM velocity in the vertical direction.

To validate our hypothesis, the input work performed by the tendon is measured in the simulation. Notably, the robot with the optimized design showed 1.53 times more input work than the base model. We hypothesize this is due to the slightly tilted shapes of the upper link structures. This

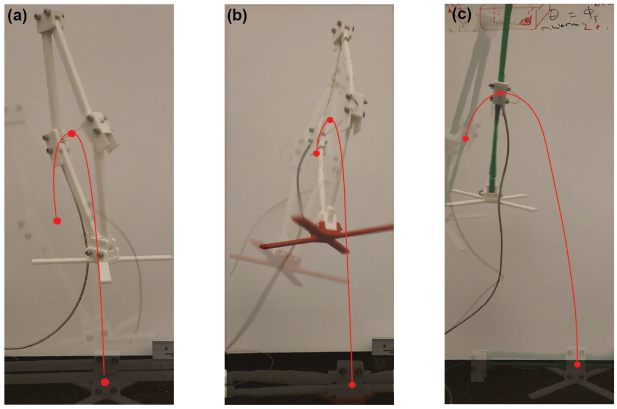

Fig. 6. **Jumping locomotion of the base and AI-generated robots. (a)** Base robot; **(b)** AI-generated robot before optimization; **(c)** AI-generated robot after optimization.

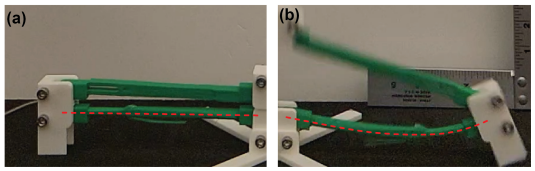

Fig. 7. **Deformation of the lower link at the jumping moment. (a)** shows the original shape of the lower link before jumping and **(b)** shows how it deforms at the jumping moment.

creates singular positions that prevent the robot from jumping until the tension reaches a certain level. This increases input work, as it is the inner product of tension and displacement. Another fact we found while operating the real-world robot is that the lower links deform (Fig. 7), which may contribute in jumping by storing and emitting elastic energy. This phenomenon is particularly intriguing, as deformation was not accounted for in the optimization process. Despite the limitations in simulation accuracy, the simulator could successfully found a creative design by finding a design that jump after accumulating sufficient input work provided from tendon-motor system.

VI. DISCUSSION AND CONCLUSION

This paper presents an AI framework for robot design that leverages a diffusion model to generate optimized robot structures. By combining AI generation with human-guided mesh operation, our framework creates robot designs without the need for post-processing. We demonstrate the effectiveness of this approach by applying it to the design of jumping robots, showing improved performance in both jumping and landing. Our solution embodies a collaborative effort between human designers and AI, with each complementing the other's strengths. Human designers excel in considering fabrication and assembly constraints due to their deeper understanding, while AI models can efficiently explore vast design spaces, iterating quickly under predefined constraints. This synergy between human intuition and AI-driven exploration accelerates the design process and has the potential to inspire new insights for human researchers.

REFERENCES

[1] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[2] T. Zhu, J. Hooks, and D. Hong, "Design, modeling, and analysis of a liquid cooled proprioceptive actuator for legged robots," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2019, pp. 36–43.

[3] B. Kim, U. Jeong, B. B. Kang, and K. J. Cho, "Slider-Tendon Linear Actuator with Under-actuation and Fast-connection for Soft Wearable Robots," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 2932–2943, 2021.

[4] M. Bauza, A. Bronars, Y. Hou, I. Taylor, N. Chavan-Dafle, and A. Rodriguez, "Simple, a visuotactile method learned in simulation to precisely pick, localize, regrasp, and place objects," *Science Robotics*, vol. 9, no. 91, p. eadi8808, 2024.

[5] Y.-M. Chen, H. Bui, and M. Posa, "Reinforcement learning for reduced-order models of legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5801–5807.

[6] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[7] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[8] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics," *science*, vol. 318, no. 5853, pp. 1088–1093, 2007.

[9] S.-M. Baek, S. Yim, S.-H. Chae, D.-Y. Lee, and K.-J. Cho, "Ladybird beetle–inspired compliant origami," *Science Robotics*, vol. 5, no. 41, p. eaaz6262, 2020.

[10] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[11] C. Laschi, J. Rossiter, F. Iida, M. Cianchetti, and L. Margheri, *Soft robotics: trends, applications and challenges*. Springer, 2017, vol. 17.

[12] A. Hassan and M. Abomoharam, "Modeling and design optimization of a robot gripper mechanism," *Robotics and Computer-Integrated Manufacturing*, vol. 46, pp. 94–103, 2017.

[13] P. Ma, T. Du, J. Z. Zhang, K. Wu, A. Spielberg, R. K. Katzschmann, and W. Matusik, "Diffaqua: A differentiable computational design pipeline for soft underwater swimmers with shape interpolation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[14] 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, 2024.

[15] 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.

[16] F. Li, W. Liu, X. Fu, G. Bonsignori, U. Scarfogliero, C. Stefanini, and P. Dario, "Jumping like an insect: Design and dynamic optimization of a jumping mini robot based on bio-mimetic inspiration," *Mechatronics*, vol. 22, no. 2, pp. 167–176, 2012.

[17] J.-S. Koh, E. Yang, G.-P. Jung, S.-P. Jung, J. H. Son, S.-I. Lee, P. G. Jablonski, R. J. Wood, H.-Y. Kim, and K.-J. Cho, "Jumping on water: Surface tension–dominated jumping of water striders and robotic insects," *Science*, vol. 349, no. 6247, pp. 517–521, 2015.

[18] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. S. Grathwohl, "Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 8489–8510.

[19] C. Zhang, W. Zou, L. Ma, and Z. Wang, "Biologically inspired jumping robots: A comprehensive review," *Robotics and Autonomous Systems*, vol. 124, p. 103362, 2020.

[20] E. W. Hawkes, C. Xiao, R.-A. Peloquin, C. Keeley, M. R. Begley, M. T. Pope, and G. Niemeyer, "Engineered jumpers overcome biological limits via work multiplication," *Nature*, vol. 604, no. 7907, pp. 657–661, 2022.

[21] G.-P. Jung, C. S. Casarez, S.-P. Jung, R. S. Fearing, and K.-J. Cho, "An integrated jumping-crawling robot using height-adjustable jumping module," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4680–4685.

[22] J. Lo and B. Parslew, "An energetic analysis of rhomboidal linkage robots for hopping lunar exploration," in *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, 2021, pp. 1–5.

[23] M. Kovac, M. Fuchs, A. Guignard, J.-C. Zufferey, and D. Floreano, "A miniature 7g jumping robot," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 373–378.

[24] G.-P. Jung, C. S. Casarez, J. Lee, S.-M. Baek, S.-J. Yim, S.-H. Chae, R. S. Fearing, and K.-J. Cho, "Jumproach: A trajectory-adjustable integrated jumping–crawling robot," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 947–958, 2019.

[25] M. Noh, S.-W. Kim, S. An, J.-S. Koh, and K.-J. Cho, "Flea-inspired catapult mechanism for miniature jumping robots," *IEEE transactions on robotics*, vol. 28, no. 5, pp. 1007–1018, 2012.

[26] V. Zaitsev, O. Gvirsman, U. B. Hanan, A. Weiss, A. Ayali, and G. Kosa, "A locust-inspired miniature jumping robot," *Bioinspiration & biomimetics*, vol. 10, no. 6, p. 066012, 2015.

[27] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[28] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[29] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5826–5835.

[30] 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.

[31] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[32] T.-H. Wang, P. Ma, A. E. Spielberg, Z. Xian, H. Zhang, J. B. Tenenbaum, D. Rus, and C. Gan, "Softzoo: A soft robot co-design benchmark for locomotion in diverse environments," *arXiv preprint arXiv:2303.09555*, 2023.

[33] H. Ha, S. Agrawal, and S. Song, "Fit2form: 3d generative model for robot gripper form design," in *Conference on Robot Learning*. PMLR, 2021, pp. 176–187.

[34] J. Hu, J. Whitman, M. Travers, and H. Choset, "Modular robot design optimization with generative adversarial networks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4282–4288.

[35] J. Hu, J. Whitman, and H. Choset, "Glso: grammar-guided latent space optimization for sample-efficient robot design automation," in *Conference on Robot Learning*. PMLR, 2023, pp. 1321–1331.

[36] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022.

[37] H. Jun and A. Nichol, "Shap-e: Generating conditional 3d implicit functions," *arXiv preprint arXiv:2305.02463*, 2023.

[38] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, "Compositional visual generation with composable diffusion models," in *European Conference on Computer Vision*. Springer, 2022, pp. 423–439.

[39] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[40] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[41] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[42] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. S. Grathwohl, "Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc," in *International conference on machine learning*. PMLR, 2023, pp. 8489–8510.

[43] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu, "Multi-concept customization of text-to-image diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1931–1941.

[44] L. John and P. Ben, "Characterising the take-off dynamics and energy efficiency in spring-driven jumping robots," *Mechanism and Machine Theory*, vol. 199, p. 105688, 2024.