
DiffuseBot: Breeding Soft Robots With Physics-Augmented Generative Diffusion Models

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Nature evolves creatures with a high complexity of morphological and behavioral
2 intelligence, meanwhile computational methods lag in approaching that diversity
3 and efficacy. Co-optimization of artificial creatures' morphology and control *in*
4 *silico* shows promise for applications in physical soft robotics and virtual character
5 creation; such approaches, however, require developing new learning algorithms
6 that can reason about function atop pure structure. In this paper, we present Diffuse-
7 Bot, a physics-augmented diffusion model that generates soft robot morphologies
8 capable of excelling in a wide spectrum of tasks. DiffuseBot bridges the gap
9 between virtually generated content and physical utility by (*i*) augmenting the
10 diffusion process with a physical dynamical simulation which provides a certificate
11 of performance, and (*ii*) introducing a co-design procedure that jointly optimizes
12 physical design and control by leveraging information about physical sensitivities
13 from differentiable simulation. We showcase a range of simulated and fabricated
14 robots along with their capabilities. Videos can be found on the anonymous website
15 <https://sites.google.com/view/diffusebot>.

16

1 Introduction

17 Designing dynamical virtual creatures or real-world cyberphysical systems requires reasoning about
18 complex trade-offs in system geometry, components, and behavior. But, what if designing such sys-
19 tems could be made simpler, or even automated wholesale from high-level functional specifications?
20 Freed to focus on higher-level tasks, engineers could explore, prototype, and iterate more quickly,
21 focusing more on understanding the problem, and find novel, more performant designs. We present
22 DiffuseBot , a first step toward efficient automatic robotic and virtual creature content creation, as
23 an attempt at closing the stubborn gap between the wide diversity and capability of Nature *vis-a-vis*
24 evolution, and the reiterative quality of modern soft robotics.

25 Specifically, we leverage diffusion-based algorithms as a means of efficiently and generatively
26 co-designing soft robot morphology and control for target tasks. Compared with previous ap-
27 proaches, DiffuseBot 's learning-based approach maintains evolutionary algorithms' ability to search
28 over diverse forms while exploiting the efficient nature of gradient-based optimization. DiffuseBot is
29 made possible by the revolutionary progress of AI-driven content generation, which is now able to
30 synthesize convincing media such as images, audio, and animations, conditioned on human input.
31 However, other than raw statistical modeling, these methods are typically task- and physics-oblivious,
32 and tend to provide no fundamental reasoning about the performance of generated outputs. We
33 provide the first method for bridging the gap between diffusion processes and the morphological
34 design of cyberphysical systems, guided by physical simulation, enabling the computational creative
35 design of virtual and physical creatures.

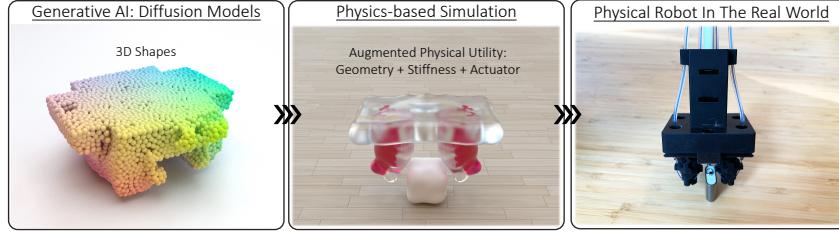


Figure 1: DiffuseBot aims to augment diffusion models with physical utility and designs for high-level functional specifications including robot geometry, material stiffness, and actuator placement.

36 While diffusion methods can robustly sample objects with coherent spatial structure from raw noise
 37 in a step-by-step fashion, several roadblocks preventing existing generative algorithms from being
 38 directly applied to physical soft robot co-design. First, while existing diffusion methods can generate
 39 2D or 3D shapes, useful for, say, sampling robot geometry, they do not consider physics, nor are
 40 they directly aligned with the robotic task performance. As an alternative approach, one might
 41 consider learning a diffusion model supervised directly on a dataset of highly-performant robot
 42 designs mapped to their task performance. This leads us to the second roadblock, that is, that no such
 43 dataset exists, and, more crucially, that curating such a dataset would require a prohibitive amount of
 44 human effort and would fail to transfer to novel tasks outside that dataset.

45 To tackle these challenges, we propose using physical simulation to guide the generative process of
 46 pretrained large-scale 3D diffusion models. Diffusion models pretrained for 3D shapes provide an
 47 expressive base distribution that can effectively propose reasonable candidate geometries for soft
 48 robots. Next, we develop an automatic procedure to convert raw 3D geometry to a representation
 49 compatible with soft body simulation, *i.e.* one that parameterizes actuator placement and specifies
 50 material stiffness. Finally, in order to sample robots in a physics-aware and performance-driven
 51 manner, we apply two methods that leverage physically based simulation. First, we optimize the
 52 embeddings that condition the diffusion model, skewing the sampling distribution toward better-
 53 performing robots as evaluated by our simulator. Second, we reformulate the sampling process that
 54 incorporates co-optimization over structure and control. We showcase the proposed approach of
 55 DiffuseBot by demonstrating automatically synthesized, novel robot designs for a wide spectrum of
 56 tasks, including balancing, landing, crawling, hurdling, gripping, and moving objects, and demonstrate
 57 its superiority to comparable approaches. We further demonstrate DiffuseBot’s amenability to
 58 incorporating human semantic input as part of the robot generation process. Finally, we demonstrate
 59 the physical realizability of the robots generated by DiffuseBot with a proof-of-concept 3D printed
 60 real-world robot, introducing the possibility of AI-powered end-to-end CAD-CAM pipelines.

61 In summary, we contribute:

- 62 • A new framework that augments the diffusion-based synthesis with physical dynamical simulation
 63 in order to generatively co-design task-driven soft robots in morphology and control.
- 64 • Methods for driving robot generation in a task-driven way toward improved physical utility by
 65 optimizing input embeddings and incorporating differentiable physics into the diffusion process.
- 66 • Extensive experiments in simulation to verify the effectiveness of DiffuseBot, extensions to text-
 67 conditioned functional robot design, and a proof-of-concept physical robot as a real-world result.

68 2 Method

69 In this section, we first formulate the problem (Section 2.1) and then describe the proposed DiffuseBot
 70 framework, which consists of a differentiable procedure that converts samples from the
 71 diffusion models into soft robots (Section 2.3), a technique to optimize embeddings conditioned
 72 by the diffusion model for improved physical utility (Section 2.4), and a reformulation of diffusion
 73 process into co-design optimization (Section 2.4).

74 2.1 Problem Formulation: Soft Robot Co-design

75 Soft robot co-design refers to a joint optimization of the morphology and control of soft robots. The
 76 morphology commonly involves robot geometry, body stiffness, and actuator placement. Control is
 77 the signal to the actuators prescribed by a given robot morphology. It can be formally defined as,

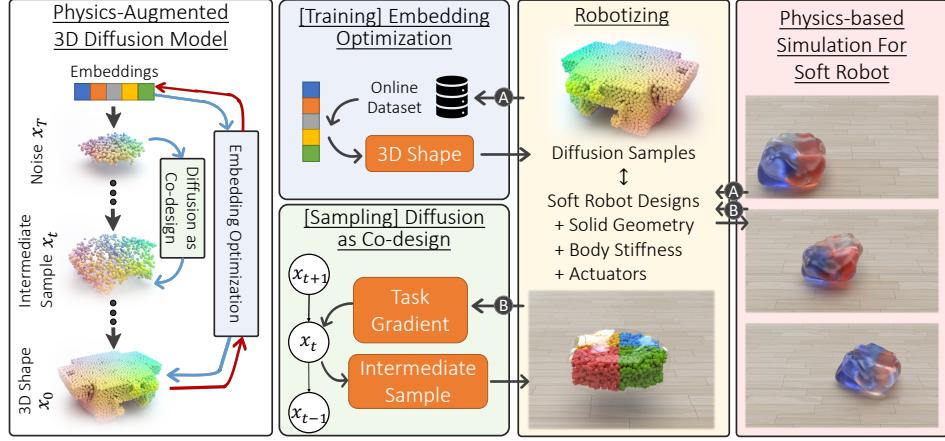


Figure 2: The DiffuseBot framework consists of three modules: (i) *robotizing*, which converts diffusion samples into physically simulatable soft robot designs (ii) *embedding optimization*, which iteratively generate new robots to be evaluated for training the conditional embedding (iii) *diffusion as co-design*, which guides the sampling process with co-design gradients from differentiable simulation. Arrow (A): evaluation of robots to guide data distribution. (B): differentiable physics as feedback.

$$\min_{\Psi, \phi} \mathcal{L}(\Psi, \phi) = \min_{\Psi, \phi} \mathcal{L}(\{\mathbf{u}_h(\mathbf{s}_h; \phi, \Psi), \mathbf{s}_h\}_{h \in [1, H]}), \quad \text{where } \mathbf{s}_{h+1} = f(\mathbf{s}_h, \mathbf{u}_h) \quad (1)$$

78 where Ψ is robot morphology that includes geometry Ψ_{geo} , stiffness Ψ_{st} , and actuator placement Ψ_{act} ;
 79 \mathbf{u}_h is actuation with the controller's parameters ϕ and dependency on the robot morphology Ψ , \mathbf{s}_h
 80 is the simulator state, f is the environmental dynamics (namely the continuum mechanics of soft
 81 robots), and H is robot time horizon (not to be confused with the later-on mentioned diffusion time).
 82 Co-design poses challenges in optimization including complex interdependencies between body
 83 and control variables, ambiguity between competing morphology modifications, trade-offs between
 84 flexibility and efficacy in design representations, etc. [48]. In this work, we aim at leveraging the
 85 generative power of diffusion models in searching for optimal robot designs with (1) the potential to
 86 synthesize highly-diverse robots and (2) inherent structural biases to achieve efficient optimization.

87 2.2 3D Shape Generation with Diffusion-based Models

88 Diffusion-based generative models [17, 40] aim to model a data distribution by augmenting
 89 it with auxiliary variables $\{\mathbf{x}_t\}_{t=1}^T$ defining a Gaussian diffusion process $p(\mathbf{x}_0) =$
 90 $\int p(\mathbf{x}_T) \prod_{t=1}^T p(\mathbf{x}_{t-1}|\mathbf{x}_t) d\mathbf{x}_{1:T}$ with the transition kernel in the forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1}) =$
 91 $\mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ for some $0 < \beta_t < 1$. For sufficiently large T , we have $p(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.
 92 This formulation enables an analytical marginal at any diffusion time $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ based
 93 on clean data \mathbf{x}_0 , where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\bar{\alpha}_t = \prod_{i=1}^t 1 - \beta_i$. The goal of the diffusion model (or
 94 more precisely the denoiser ϵ_θ) is to learn the reverse diffusion process $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ with the loss,

$$\min_{\theta} \mathbb{E}_{t \sim [1, T], p(\mathbf{x}_0), \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I})} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon, t), t)\|^2] \quad (2)$$

95 Intuitively, ϵ_θ learns a one-step denoising process that can be used iteratively during sampling to
 96 convert random noise $p(\mathbf{x}_T)$ gradually into realistic data $p(\mathbf{x}_0)$. To achieve controllable generation
 97 with conditioning \mathbf{c} , the denoising process can be slightly altered via classifier-free guidance [18, 9],

$$\hat{\epsilon}_{\theta, \text{classifier-free}} := \epsilon_\theta(\mathbf{x}_t, t, \emptyset) + s \cdot (\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) - \epsilon_\theta(\mathbf{x}_t, t, \emptyset)) \quad (3)$$

98 where s is the guidance scale, \emptyset is a null vector that represents non-conditioning.

99 2.3 Robotizing 3D Shapes from Diffusion Samples

100 We adopt Point-E [28] as a pre-trained diffusion model that is capable of generating diverse and
 101 complex 3D shapes, providing a good prior of soft robot geometries. However, the samples from the
 102 diffusion model \mathbf{x}_t are in the form of surface point cloud and are not readily usable as robots to be
 103 evaluated in the physics-based simulation. Here, we describe how to robotize the diffusion samples
 104 $\mathbf{x}_t \mapsto \Psi$ and its gradient computation of the objective $\frac{d\mathcal{L}}{d\mathbf{x}_t} = \frac{\partial \mathcal{L}}{\partial \Psi_{\text{geo}}} \frac{\partial \Psi_{\text{geo}}}{\partial \mathbf{x}_t} + \frac{\partial \mathcal{L}}{\partial \Psi_{\text{st}}} \frac{\partial \Psi_{\text{st}}}{\partial \mathbf{x}_t} + \frac{\partial \mathcal{L}}{\partial \Psi_{\text{act}}} \frac{\partial \Psi_{\text{act}}}{\partial \mathbf{x}_t}$.

Algorithm 1 Training: Embedding Optimization

Initialize: $\mathcal{D} \leftarrow \emptyset, \mathbf{c} \leftarrow \emptyset$
while within maximal number of epochs **do**
 Generate data with the diffusion model: $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0 | \mathbf{c})$.
 Evaluate samples with physics-based simulation: $l(\mathbf{x}_0) = \mathcal{L}(\Psi(\mathbf{x}_0), \phi)$.
 Aggregate and update datasets: $\mathcal{D} \leftarrow \text{Filter}(\mathcal{D} \cup \{\mathbf{x}_0, l\})$.
 Optimize the embedding \mathbf{c} on \mathcal{D} using the objective (5).
end while

105 **Solid Geometry.** We use a Material Point Method (MPM)-based simulation [48], which takes solid
106 geometries as inputs. This poses two obstacles: (1) conversion from surface point clouds into solid
107 geometries, and (2) the unstructuredness of data in the intermediate samples $\mathbf{x}_t, t \neq 0$. The second
108 issue arises from the fact that the diffusion process at intermediate steps may produce 3D points that
109 do not form a tight surface. First, we leverage the predicted clean sample at each diffusion time t ,

$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}} \quad (4)$$

This approach is used in denoising diffusion implicit model (DDIM) [41] to approximate the unknown clean sample \mathbf{x}_0 in the reverse process $p(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_0)$. Here, we use it to construct a better-structured data for simulation. Hence, we break down the robotizing process into $\mathbf{x}_t \mapsto \hat{\mathbf{x}}_0 \mapsto \Psi$ with gradient components as $\frac{\partial \Psi}{\partial \hat{\mathbf{x}}_0} \frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t}$, where $\frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t}$ can be trivially derived from (4). To convert the predicted surface points $\hat{\mathbf{x}}_0$ into solid geometry, we first reconstruct a surface mesh from $\hat{\mathbf{x}}_0$, and then evenly sample a solid point cloud Ψ_{geo} within its interior. For mesh reconstruction, we modify the optimization approach from Shape As Points [29], which provides a differentiable Poisson surface reconstruction that maps a control point set \mathbf{V}_{ctrl} to a reconstructed surface mesh with vertices \mathbf{V}_{mesh} . We calculate a modified Chamfer Distance loss indicating similarity between \mathbf{V}_{mesh} and $\hat{\mathbf{x}}_0$:

$$\mathcal{L}_{\text{recon}} = \frac{\lambda_{\text{mesh}}}{|\mathbf{V}_{\text{mesh}}|} \sum_{\mathbf{v} \in \mathbf{V}_{\text{mesh}}} d(\mathbf{v}, \hat{\mathbf{x}}_0) + \frac{\lambda_{\text{target}}}{|\hat{\mathbf{x}}_0|} \sum_{\mathbf{v} \in \hat{\mathbf{x}}_0} w(\mathbf{v}) d(\mathbf{v}, \mathbf{V}_{\text{mesh}}),$$

in which $d(\cdot, \cdot)$ denotes minimal Euclidean distance between a point and a point set, and $w(\mathbf{v})$ denotes a soft interior mask, with $w(\mathbf{v}) = 1$ for \mathbf{v} outside the mesh, and $w(\mathbf{v}) = 0.1$ for \mathbf{v} inside. The introduced mask term $w(v)$ aims to lower the influence of noisy points inside the mesh, which is caused by imperfect prediction of $\hat{\mathbf{x}}_0$ from noisy intermediate diffusion samples. The weight parameters are set to $\lambda_{\text{mesh}} = 1$ and $\lambda_{\text{target}} = 10$. We back-propagate $\frac{\partial \mathcal{L}_{\text{recon}}}{\partial \mathbf{V}_{\text{mesh}}}$ to $\frac{\partial \mathcal{L}_{\text{recon}}}{\partial \mathbf{V}_{\text{ctrl}}}$ through the differentiable Poisson solver, and then apply an Adam optimizer on \mathbf{V}_{ctrl} to optimize the loss $\mathcal{L}_{\text{recon}}$. After mesh reconstruction, the solid geometry Ψ_{geo} represented by a solid interior point cloud is then sampled evenly within the mesh, with sufficient density to support the MPM-based simulation. Finally, to integrate the solidification process into diffusion samplers, we still need its gradient $\frac{\partial \Psi_{\text{geo}}}{\partial \hat{\mathbf{x}}_0}$. We adopt Gaussian kernels on point-wise Euclidean distances as gradients between two point clouds:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{v}} = \frac{\exp(-\alpha \|\mathbf{u} - \mathbf{v}\|^2)}{\sum_{v' \in \hat{\mathbf{x}}_0} \exp(-\alpha \|\mathbf{u} - \mathbf{v}'\|^2)}, \mathbf{u} \in \Psi_{\text{geo}}, \mathbf{v} \in \hat{\mathbf{x}}_0.$$

110 Intuitively, under such Gaussian kernels gradients, each solid point is linearly controlled by predicted
111 surface points near it. In practice, this backward scheme works well for kernel parameter $\alpha = 20$.

112 **Actuators and Stiffness.** A solid geometry does not make a robot; in order for the robot to behave,
113 its dynamics must be defined. After sampling a solid geometry, we thus need to define material
114 properties and actuator placement. Specifically, we embed actuators in the robot body in the form
115 of muscle fibers that can contract or expand to create deformation; further, we define a stiffness
116 parameterization in order to determine the relationship between deformation and restorative elastic
117 force. We adopt constant stiffness for simplicity since it has been shown to trade off with actuation
118 strength [48]; thus we have $\Psi_{\text{st}}(\hat{\mathbf{x}}_0) = \text{const}$ and $\frac{\partial \Psi_{\text{st}}}{\partial \hat{\mathbf{x}}_0} = 0$. Then, we propose to construct actuators
119 based on the robot solid geometry $\Psi_{\text{act}}(\Psi_{\text{geo}}(\hat{\mathbf{x}}_0))$ via clustering; namely, we perform k-means with
120 pre-defined number of clusters on the coordinates of 3D points from the solid geometry Ψ_{geo} . The
121 gradient then becomes $\frac{\partial \Psi_{\text{act}}}{\partial \Psi_{\text{geo}}} \frac{\partial \Psi_{\text{geo}}}{\partial \hat{\mathbf{x}}_0} \frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t}$, where $\frac{\partial \Psi_{\text{act}}}{\partial \Psi_{\text{geo}}} \approx 0$ as empirically we found the clustering is
122 quite stable in terms of label assignment, i.e., with $\Delta \Psi_{\text{geo}}$ being small, $\Delta \Psi_{\text{act}} \rightarrow 0$. Overall, we keep
123 only the gradient for the robot geometry as empirically it suffices.

Algorithm 2 Sampling: Diffusion As Co-design

```

Initialize: initial sample  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
while within maximal number of diffusion steps  $t \geq 0$  do
    Perform regular per-step diffusion update:  $\mathbf{x}_t \leftarrow \mathbf{x}_{t+1}$ .
    if perform co-design then
        while within  $K$  steps do
            Run update in (6) and (7) to overwrite  $\mathbf{x}_t$ .
        end while
    end if
end while

```

124 **2.4 Physics Augmented Diffusion Model**

125 **Embedding Optimization.** To best leverage the diversity of the generation from a large-scale
126 pre-trained diffusion models, we propose to (1) actively generate new data from model and maintain
127 them in a buffer, (2) use physics-based simulation as a certificate of performance, (3) optimize
128 the embeddings conditioned by the diffusion model under a skewed data distribution to improve
129 robotic performance in simulation. Curating a training dataset on its own alleviates the burden of
130 manual effort to propose performant robot designs. Optimizing the conditional embeddings instead
131 of finetuning the diffusion model eliminates the risk of deteriorating the overall generation and saves
132 the cost of storing model weights for each new task (especially with large models). We follow,

$$\min_{\mathbf{c}} \mathbb{E}_{t \sim [1, T], p_{\theta}(\mathbf{x}_0 | \mathbf{c}), \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I})} [\|\epsilon - \epsilon_{\theta}(\mathbf{x}_t(\mathbf{x}_0, \epsilon, t), t, \mathbf{c})\|^2] \quad (5)$$

133 Note the three major distinctions from (2): (i) the optimization variable is \mathbf{c} not θ (ii) the denoiser
134 is conditioned on the embeddings $\epsilon_{\theta}(\dots, \mathbf{c})$ and (iii) the data distribution is based on the diffusion
135 model p_{θ} not the inaccessible real data distribution p and is conditioned on the embeddings \mathbf{c} . This
136 adopts an online learning scheme as the sampling distribution is dependent on the changing \mathbf{c} . The
137 procedure is briefly summarized in Algorithm 1, where *Filter* is an operation to drop the oldest data
138 when exceeding the buffer limit. In addition, during this stage, we use fixed prescribed controllers
139 since we found empirically that a randomly initialized controller may not be sufficiently informative to
140 drive the convergence toward reasonably good solutions; also, enabling the controller to be trainable
141 makes the optimization prohibitively slow and extremely unstable, potentially due to the difficulty
142 of the controller required to be universal to a diverse set of robot designs. After the embedding
143 optimization, we can perform conditional generation that synthesizes samples corresponding to robot
144 designs with improved physical utility via classifier-free guidance as in (3).

145 **Diffusion as Co-design.** While the optimized embedding already allows us to generate performant
146 robots for some target tasks, we further improve the performance of individual samples by reformulating
147 the diffusion sampling process into a co-design optimization. As described in Section 2.3, we can
148 convert the intermediate sample at any diffusion time \mathbf{x}_t to a robot design Ψ , *rendering an evolving*
149 *robot design throughout the diffusion process*. However, regular diffusion update [17] much less
150 resembles any gradient-based optimization techniques, which are shown to be effective in soft robot
151 design and control with differentiable simulation [19, 2]. Fortunately, there is a synergy between
152 diffusion models and energy-based models [42, 12, 11], which allows a more gradient-descent-like
153 update with Markov Chain Monte Carlo (MCMC) sampling. Incorporating the soft robot co-design
154 optimization with differentiable physics [48] into the diffusion sampling process, we have

$$\text{Design Optim.: } \mathbf{x}_t^{(k)} = \mathbf{x}_t^{(k-1)} + \frac{\sigma^2}{2} \left(\epsilon_{\theta}(\mathbf{x}_t^{(k-1)}, t) - \kappa \nabla_{\mathbf{x}_t^{(k-1)}} \mathcal{L}(\Psi(\mathbf{x}_t^{(k-1)}), \phi_t^{k-1}) \right) + \sigma^2 \epsilon \quad (6)$$

$$\text{Control Optim.: } \phi_t^{(k)} = \phi_t^{(k-1)} + \gamma \nabla_{\phi_t^{(k-1)}} \mathcal{L}(\Psi(\mathbf{x}_t^{(k-1)}), \phi_t^{k-1}) \quad (7)$$

155 where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_t^{(0)} = \mathbf{x}_{t-1}^{(K)}$, κ is the ratio between two types of design gradients, K is the
156 number of MCMC sampling steps at the current diffusion time, γ is the weight for trading off design
157 and control optimization, and $\phi_t^{(0)}$ can be either inherited from the previous diffusion time $\phi_{t-1}^{(K)}$ or
158 reset to the initialization $\phi_T^{(0)}$. We highlight the high resemblance to gradient-based co-optimization
159 with \mathbf{x}_t as the design variable and ϕ_t as the control variable. This procedure is performed once every
160 M diffusion steps (Algorithm 2), where M is a hyperparameter that trade-offs "guidance" strength



Figure 3: We consider passive dynamics tasks (balancing, landing), locomotion tasks (crawling, hurdling), and manipulation tasks (gripping, moving a box).

Table 1: Improved physical utility by augmenting physical simulation with diffusion models.

Embed. Optim.	Diffusion as Co-design	Passive Dynamics		Locomotion		Manipulation	
		Balancing	Landing	Crawling	Hurdling	Gripping	Moving a Box
✓		0.081 ^{.164}	0.832 ^{.217}	0.011 ^{.012}	0.014 ^{.020}	0.014 ^{.008}	0.019 ^{.020}
✓	✓	0.556 ^{.127}	0.955 ^{.032}	0.048 ^{.007}	0.019 ^{.014}	0.025 ^{.006}	0.040 ^{.018}
		0.653^{.107}	0.964^{.029}	0.081^{.018}	0.035^{.030}	0.027^{.004}	0.044^{.021}

from physical utility and sampling efficiency. Intuitively, the entire diffusion-as-co-design process is guided by three types of gradients: (i) $\epsilon_\theta(\mathbf{x}_t^{(k-1)}, \cdot)$ provides a direction for the design toward feasible 3D shapes based on the knowledge of pre-training with large-scale datasets (and toward enhanced physical utility with the optimized embeddings via classifier-free guidance using $\hat{\epsilon}_\theta(\mathbf{x}_t^{(k-1)}, \cdot, \mathbf{c})$), (ii) $\nabla_{\mathbf{x}_t} \mathcal{L}(\Psi(\mathbf{x}_t), \cdot)$ provides a direction for the design toward improving co-design objective \mathcal{L} via differentiable simulation, and (iii) $\nabla_{\phi_t} \mathcal{L}(\cdot, \phi_t)$ provides a direction for the controller toward a better adaption to the current design \mathbf{x}_t that allows more accurate evaluation of the robot performance.

3 Experiments

3.1 Task Setup

We cover three types of robotics tasks: passive dynamics, locomotion, and manipulation (Figure 3).

- **Passive Dynamics** tasks balancing and targeted landing. *Balancing* initializes the robot on a stick-like platform with small contact area with an upward velocity that introduces instability; the goal is for the robot to passively balance itself after dropping on the platform. *Landing* applies an initial force to toss the robot toward a target object; the goal is for the robot to passively land as close as to the target as possible.
- **Locomotion** tasks include crawling and hurdling. *Crawling* sets the robot at a rest state on the ground; the robot must actuate its body to move as far away as possible from the starting position. *Hurdling* places an obstacle in front of the robot; the robot must jump over the obstacle.
- **Manipulation** tasks include gripping and moving objects. *Gripping* places an object underneath the robot; the goal of the robot is to vertically lift the object. *Box Moving* places a box on the right end of the robot; the robot must move the box to the left.

We refer the reader to the appendix for more detailed descriptions.

3.2 Toward Physical Utility In Diffusion Models

Physics-augmented diffusion. In Table 1, we examine the effectiveness of embedding optimization and diffusion as co-design for improving physical utility. For each entry, we draw 100 samples with preset random seeds to provide valid sample-level comparison (i.e., setting the step size of co-design optimization to zero in the third row will produce almost identical samples as the second row). We report the average performance with standard deviation in the superscript. First, we observe increasing performance across all tasks while incorporating the two proposed techniques, demonstrating the efficacy of DiffuseBot. Besides, the sample-level performance does not always monotonically improve, possibly due to the stochasticity within the diffusion process and the low quality of gradient from differentiable simulation in some scenarios. For example, in gripping, when the robot fails to pick up the object in the first place, the gradient may be informative and fails to bring proper guidance toward better task performance; similarly in moving a box. In addition, we found it necessary to include control optimization during the diffusion sampling process, since, at diffusion steps further from zero, the predicted clean sample $\hat{\mathbf{x}}_0$ (derived from the intermediate sample \mathbf{x}_t) may differ significantly from the clean sample \mathbf{x}_0 , leaving the prescribed controller largely unaligned.

Table 2: Comparison with baselines.

Methods	Passive Dynamics		Locomotion		Manipulation	
	Balancing	Landing	Crawling	Hurdling	Gripping	Moving a Box
Particle-based	0.040	0.870	0.022	0.007	-0.007	0.083
Voxel-based	0.040	0.856	0.024	0.027	-0.009	0.078
Implicit Function [26]	0.682	0.949	0.084	0.149	0.030	0.087
Diff-CPPN [14]	0.287	0.954	0.081	0.109	0.024	0.057
DiffuseBot (Ours)	0.919	0.998	0.115	0.178	0.031	0.091



Figure 4: Examples of DiffuseBot evolving robots to solve different tasks.

198 **Comparison with baselines.** In Table 2, we compare with extensive baselines of soft robot design
 199 representation. For each baseline method, we run the co-optimization routine for the same number
 200 of steps as in the diffusion-as-co-design stage in DiffuseBot. To avoid being trapped in the local
 201 optimum, we run each baseline with 20 different random initializations and choose the best one.
 202 Since DiffuseBot is a generative method, we draw 20 samples and report the best; this is sensible an
 203 applications-driven perspective since we only need to retrieve one performant robot, within a reason-
 204 able sample budget. We observe that our method outperforms all baselines. DiffuseBot leverages the
 205 knowledge of large-scale pre-trained models that capture the “common sense” of geometry, providing
 206 a more well-structured yet flexible prior for soft robot design.

207 **Soft robots bred by DiffuseBot.** In Figure 5, we demonstrate the generated soft robots that excel
 208 in locomotion and manipulation tasks. We highlight the flexibility of DiffuseBot to generate highly
 209 diverse soft robot designs that accommodate various purposes in different robotics tasks. Furthermore,
 210 in Figure 4, we show how robots evolve from a feasible yet non-necessarily functional design to
 211 an improved one that intuitively matches the task objective. By manually inspecting the evolving
 212 designs, we found that the role of the embedding optimization is to drive the diverse generations
 213 toward a converged, smaller set with elements having higher chance to succeed the task; on the other
 214 hand, the role of diffusion as co-design brings relatively minor tweaks along with alignment between
 215 the control and design. Due to space limit, we refer the reader to our project page for more results.

216 3.3 Ablation Analysis

217 In this section, we conduct a series of ablation studies to provide a deeper understanding of the
 218 proposed method. For simplicity, all experiments in this section are done with the crawling task.

Table 3: Ablation on embedding optimization. MT means manually-designed text. FT means finetuning models.

	Performance
MT	0.016 ^{-0.14}
FT	0.031 ^{-0.24}
Ours	0.048 ^{-0.07}

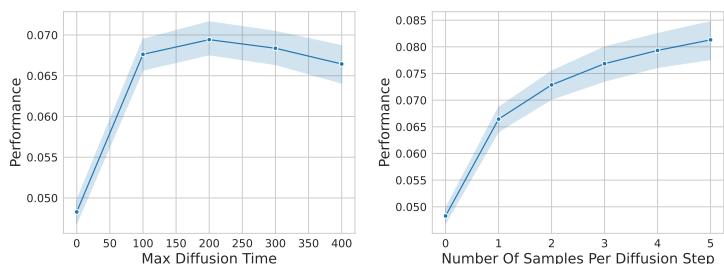


Table 4: Varying starting point and strength of diffusion as co-design.

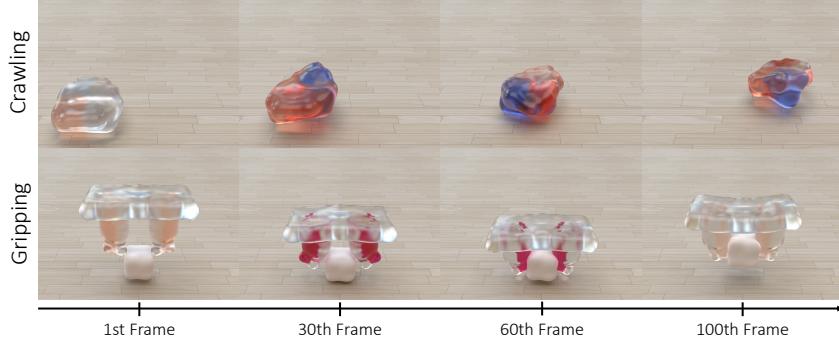


Figure 5: Examples of robots bred by DiffuseBot to achieve the desired tasks.

219 **Embedding optimization.** In Table 3, we compare the optimization of the embedding conditioned by
 220 the diffusion models with other alternatives. The pre-trained diffusion model [28] that DiffuseBot is
 221 built upon uses CLIP embeddings [32], which allows for textual inputs. Hence, a naive approach is
 222 to manually design text for the conditional embedding of the diffusion model. The result reported
 223 in Table 3 uses “*a legged animal or object that can crawl or run fast*”. We investigated the use of
 224 text prompts; in our experience, text was difficult to optimize for *functional* robot design purposes.
 225 This is expected since most existing diffusion models perform content generation only in terms of
 226 appearance instead of physical utility, which further strengthens the purpose of this work. In addition,
 227 with exactly the same training objective as in (5), we can instead finetune the diffusion model itself.
 228 However, this does not yield better performance, as shown in the second entry in Table 3. Empirically,
 229 we found there is a higher chance of the generated samples being non-well-structured with fractured
 230 parts. This suggests that finetuning for physical utility may deteriorate the modeling of sensible 3D
 231 shapes and lead to more unstable generations.

232 **Diffusion as co-design.** Recall that the co-design optimization can be seamlessly incorporated into
 233 any diffusion step. In Figure 4, we examine how the strength of the injected co-design optimization
 234 affects the task performance in terms of where to apply throughout the diffusion sampling process
 235 and how many times to apply. In the left figure of Figure 4, we sweep through the maximal diffusion
 236 time of applying diffusion as co-design, i.e., for the data point at $t = 400$, we only perform co-design
 237 from $t = 400$ to $t = 0$. We found that there is a sweet spot of when to start applying co-design
 238 (at $t \approx 200$). This is because the intermediate samples at larger diffusion time $x_t, t \gg 0$ are
 239 extremely under-developed, lacking sufficient connection to the final clean sample x_0 , hence failing
 240 to provide informative guidance by examining its physical utility. Furthermore, we compare against
 241 post-diffusion co-design optimization, i.e., run co-design based on the final output of the diffusion
 242 (0.064 vs ours 0.081). We allow the same computational budget by running the same number of
 243 times of differentiable simulation as in DiffuseBot. Our method performs slightly better, potentially
 244 due to the flexibility to alter the still-developing diffusion samples. Also note that while our method is
 245 interleaved into diffusion process, it is still compatible with any post-hoc computation for finetuning.

246 3.4 Flexibility To Incorporate Human Feedback

247 Beyond the strong generative power, diffusion
 248 models also provide the flexibility to
 249 composite different data distributions. This
 250 is especially useful for computational de-
 251 sign since it empowers to easily incorporate
 252 external knowledge, e.g. from human. We
 253 follow the compositionality techniques in-
 254 troduced in [24, 11], which can be directly
 255 integrated into our diffusion as co-design
 256 framework. In Figure 6, we demonstrate incorpo-
 257 rating human feedback in textual form as “*a unicorn*”
 into a crawling robot generated by DiffuseBot. We can see the emergence of the horn-like body part.

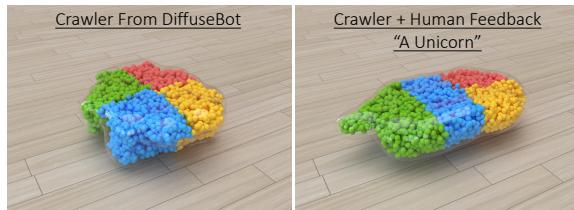


Figure 6: Incorporating human textual feedback.

258 3.5 From Virtual Generation To Physical Robot

259 We further fabricate a physical robot for the gripping task as a proof-of-concept to demonstrate the
 260 possibility of real-world extension. We use a 3D Carbon printer to reconstruct the exact geometry
 261 of a generated design and fill the robot body with a Voronoi lattice structure to achieve softness.

262 For actuators, we employ tendon transmission to realize the contraction force utilized in the soft
263 robot gripper. In our project page, we demonstrate the robots generated by DiffuseBot are capable
264 of picking up an object. Note that physical robot fabrication and real-world transfer have countless
265 non-trivial challenges including stiffness and actuator design, sim-to-real gap, etc. Hence, this
266 experiment is only meant to demonstrate the potential instead of a general, robust pipeline toward
267 physical robots, which is left to future work. We refer the reader to the appendix for more details.

268 4 Related Work

269 **Heuristic Search For Soft Robot Co-Design.** Heuristic searches are simple but useful tools for co-
270 designing soft robots. A long line of work has focused on evolutionary algorithms [4, 5, 7], with some
271 including physical demonstrations [15, 21, 22] and recent benchmarks incorporating neural control
272 [3]. These methods are often powered by parameterized by compositional pattern-producing networks
273 [46], which parameterize highly expressive search spaces akin to neural networks [38, 39]. Similar
274 to [3], [35] combines a heuristic approach with reinforcement learning, and demonstrates resulting
275 designs on physical hardware. Other notable methods include particle-filter-based approaches [8] and
276 simulated annealing [47]. Heuristic search methods tend to be less efficient than gradient-based or
277 learning-based algorithms, but can reasoning about large search spaces; our approach employs the
278 highly expressive diffusion processes, while leveraging the differentiable nature of neural networks
279 and physical simulation for more efficient and gradient-directed search.

280 **Gradient-Based Soft Robot Co-Optimization.** A differentiable simulator is one in which useful
281 analytical derivatives of any system variable with respect to any other system variable is efficiently
282 queryable; the recent advent of soft differentiable simulation environments [19, 44, 10, 23, 30, 31, 48]
283 has accelerated the exploration of gradient-based co-optimization methods. [19, 44] demonstrated
284 how differentiable simulators can be used to co-optimize very high-dimensional spatially varying
285 material and open-loop/neural controller parameters. [27] presented gradient-based search of shape
286 parameters for soft manipulators. Meanwhile, [48] showed how actuation and geometry could be
287 co-optimized, while analyzing the trade-offs of design space complexity and exploration in the search
288 procedure. DiffuseBot borrows ideas from gradient-based optimization in guiding the design search
289 in a physics-aware way, especially in the context of control.

290 **Learning-Based Soft Robot Co-Design Methods.** Though relatively nascent, learning-based ap-
291 proaches (including DiffuseBot) can re-use design samples to build knowledge about a problem.
292 Further, dataset-based minibatch optimization algorithms are more robust to local minima than
293 single-iterate pure optimization approaches. [45] demonstrated how gradient-based search could be
294 combined with learned-models; a soft robot proprioceptive model was continually updated by simu-
295 lation data from interleaved control/material co-optimization. Other work employed learning-based
296 methods in the context of leveraging available datasets. [25] learned a parameterized representation
297 of geometry and actuators from basis shape geometries tractable interpolation over high-dimensional
298 search spaces. [43] leveraged motion data and sparsifying neurons to simultaneously learn sensor
299 placement and neural soft robotic tasks such as proprioception and grasp classification.

300 **Diffusion Models for Content Generation.** Diffusion models [17, 40] have emerged as the de-facto
301 standard for generating content in continuous domains such as images [33, 34], 3D content [53, 52],
302 controls [20, 6, 1], videos [16, 37, 13], and materials [50, 49, 36]. In this paper, we explore how
303 diffusion models in combination with differentiable physics may be used to design new robots. Most
304 similar to our work, [51] uses differentiable physics to help guide human motion synthesis. However,
305 while [51] uses differentiable physics to refine motions in the last few timesteps of diffusion sampling,
306 we tightly integrate differentiable physics wih sampling throughout the diffusion sampling procedure
307 through MCMC. We further uses differentiable simulation to define a reward objective through which
308 we may optimize generative embeddings that represent our desirable robot structure.

309 5 Conclusion

310 We presented DiffuseBot, a framework that augments physics-based simulation with a diffusion
311 process capable of generating performant soft robots for a diverse set of tasks including passive
312 dynamics, locomotion, and manipulation. We demonstrated the efficacy of diffusion-based generation
313 with extensive experiments, presented a method for incorporating human feedback, and prototyped a
314 physical robot counterpart. DiffuseBot is a first step toward generative invention of soft machines,
315 with the potential to accelerate design cycles, discover novel devices, and provide building blocks for
316 downstream applications in automated computational creativity and computer-assisted design.

317 **References**

- 318 [1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit
319 Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint*
320 *arXiv:2211.15657*, 2022.
- 321 [2] Moritz Bächer, Espen Knoop, and Christian Schumacher. Design and control of soft robots
322 using differentiable simulation. *Current Robotics Reports*, 2(2):211–221, 2021.
- 323 [3] Jagdeep Bhatia, Holly Jackson, Yunsheng Tian, Jie Xu, and Wojciech Matusik. Evolution gym:
324 A large-scale benchmark for evolving soft robots. *Advances in Neural Information Processing*
325 *Systems*, 34:2201–2214, 2021.
- 326 [4] Nicholas Cheney, Jeff Clune, and Hod Lipson. Evolved electrophysiological soft robots. In
327 *Artificial life conference proceedings*, pages 222–229. MIT Press One Rogers Street, Cambridge,
328 MA 02142-1209, USA journals-info . . . , 2014.
- 329 [5] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving
330 soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution*,
331 7(1):11–23, 2014.
- 332 [6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and
333 Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *arXiv*
334 *preprint arXiv:2303.04137*, 2023.
- 335 [7] Francesco Corucci, Nick Cheney, Francesco Giorgio-Serchi, Josh Bongard, and Cecilia Laschi.
336 Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties
337 and environmental transitions. *Soft robotics*, 5(4):475–495, 2018.
- 338 [8] Raphael Deimel, Patrick Irmisch, Vincent Wall, and Oliver Brock. Automated co-design of soft
339 hand morphology and control strategy for grasping. In *2017 IEEE/RSJ International Conference*
340 *on Intelligent Robots and Systems (IROS)*, pages 1213–1218. IEEE, 2017.
- 341 [9] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image
342 synthesis. In *Advances in Neural Information Processing Systems*, 2021.
- 343 [10] Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech
344 Matusik. Diffpd: Differentiable projective dynamics. *ACM Transactions on Graphics (TOG)*,
345 41(2):1–21, 2021.
- 346 [11] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus,
347 Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle:
348 Compositional generation with energy-based diffusion models and mcmc. *arXiv preprint*
349 *arXiv:2302.11552*, 2023.
- 350 [12] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models.
351 *arXiv preprint arXiv:1903.08689*, 2019.
- 352 [13] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B Tenenbaum, Dale
353 Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation.
354 *arXiv e-prints*, pages arXiv–2302, 2023.
- 355 [14] Chrisantha Fernando, Dylan Banarse, Malcolm Reynolds, Frederic Besse, David Pfau, Max
356 Jaderberg, Marc Lanctot, and Daan Wierstra. Convolution by evolution: Differentiable pattern
357 producing networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*
358 2016, pages 109–116, 2016.
- 359 [15] Jonathan Hiller and Hod Lipson. Automatic design and manufacture of soft robots. *IEEE*
360 *Transactions on Robotics*, 28(2):457–466, 2011.
- 361 [16] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko,
362 Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen
363 video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*,
364 2022.

- 365 [17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances
366 in Neural Information Processing Systems*, 33:6840–6851, 2020.
- 367 [18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint
368 arXiv:2207.12598*, 2022.
- 369 [19] Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B Tenenbaum, William T Freeman,
370 Jiajun Wu, Daniela Rus, and Wojciech Matusik. Chainqueen: A real-time differentiable physical
371 simulator for soft robotics. In *2019 International conference on robotics and automation (ICRA)*,
372 pages 6265–6271. IEEE, 2019.
- 373 [20] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for
374 flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.
- 375 [21] Sam Kriegman, Douglas Blackiston, Michael Levin, and Josh Bongard. A scalable pipeline
376 for designing reconfigurable organisms. *Proceedings of the National Academy of Sciences*,
377 117(4):1853–1859, 2020.
- 378 [22] Sam Kriegman, Douglas Blackiston, Michael Levin, and Josh Bongard. Kinematic self-
379 replication in reconfigurable organisms. *Proceedings of the National Academy of Sciences*,
380 118(49):e2112672118, 2021.
- 381 [23] Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. Diffcloth: Differentiable cloth
382 simulation with dry frictional contact. *ACM Transactions on Graphics (TOG)*, 42(1):1–20,
383 2022.
- 384 [24] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional
385 visual generation with composable diffusion models. *arXiv preprint arXiv:2206.01714*, 2022.
- 386 [25] Pingchuan Ma, Tao Du, John Z Zhang, Kui Wu, Andrew Spielberg, Robert K Katzschmann, and
387 Wojciech Matusik. Diffqua: A differentiable computational design pipeline for soft underwater
388 swimmers with shape interpolation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.
- 389 [26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger.
390 Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the
391 IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.
- 392 [27] Thomas Morzadec, Damien Marcha, and Christian Duriez. Toward shape optimization of soft
393 robots. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 521–526.
394 IEEE, 2019.
- 395 [28] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A
396 system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*,
397 2022.
- 398 [29] Songyou Peng, Chiyu "Max" Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas
399 Geiger. Shape as points: A differentiable poisson solver. In *Advances in Neural Information
400 Processing Systems (NeurIPS)*, 2021.
- 401 [30] Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming C Lin. Scalable differentiable physics
402 for learning and control. *arXiv preprint arXiv:2007.02168*, 2020.
- 403 [31] Yiling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. Differentiable simulation of soft
404 multi-body systems. *Advances in Neural Information Processing Systems*, 34:17123–17135,
405 2021.
- 406 [32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
407 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
408 models from natural language supervision. In *International conference on machine learning*,
409 pages 8748–8763. PMLR, 2021.
- 410 [33] Aditya Ramesh, Mikhail Pavlov, Scott Gray Gabriel Goh, Chelsea Voss, Alec Radford, Mark
411 Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*,
412 2021.

- 413 [34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed
 414 Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim
 415 Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image
 416 diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.
- 417 [35] Charles Schaff, Audrey Sedal, and Matthew R Walter. Soft robots learn to crawl: Jointly
 418 optimizing design and control with sim-to-real transfer. *arXiv preprint arXiv:2202.04575*, 2022.
- 419 [36] Arne Schnieuing, Yuanqi Du, Arian Jamasb Charles Harris, Ilia Igashov, Weitao Du, Tom
 420 Blundell, Pietro Lió, Carla Gomes, Michael Bronstein Max Welling, and Bruno Correia.
 421 Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.02303*,
 422 2022.
- 423 [37] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu,
 424 Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without
 425 text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- 426 [38] Lawrence Smith, Travis Hainsworth, Jacob Haimes, and Robert MacCurdy. Automated synthesis
 427 of bending pneumatic soft actuators. In *2022 IEEE 5th International Conference on Soft Robotics
 (RoboSoft)*, pages 358–363. IEEE, 2022.
- 429 [39] Lawrence Smith and Robert MacCurdy. Soroforge: End-to-end soft actuator design. *IEEE
 430 Transactions on Automation Science and Engineering*, 2023.
- 431 [40] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsuper-
 432 vised learning using nonequilibrium thermodynamics. In *International Conference on Machine
 433 Learning*, pages 2256–2265. PMLR, 2015.
- 434 [41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv
 435 preprint arXiv:2010.02502*, 2020.
- 436 [42] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data
 437 distribution. *Advances in neural information processing systems*, 32, 2019.
- 438 [43] Andrew Spielberg, Alexander Amini, Lillian Chin, Wojciech Matusik, and Daniela Rus. Co-
 439 learning of task and sensor placement for soft robotics. *IEEE Robotics and Automation Letters*,
 440 6(2):1208–1215, 2021.
- 441 [44] Andrew Spielberg, Tao Du, Yuanming Hu, Daniela Rus, and Wojciech Matusik. Advanced soft
 442 robot modeling in chainqueen. *Robotica*, 41(1):74–104, 2023.
- 443 [45] Andrew Spielberg, Allan Zhao, Yuanming Hu, Tao Du, Wojciech Matusik, and Daniela Rus.
 444 Learning-in-the-loop optimization: End-to-end control and co-design of soft robots through
 445 learned deep latent representations. *Advances in Neural Information Processing Systems*, 32,
 446 2019.
- 447 [46] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of develop-
 448 ment. *Genetic programming and evolvable machines*, 8:131–162, 2007.
- 449 [47] Merel Van Diepen and Kristina Shea. A spatial grammar method for the computational design
 450 synthesis of virtual soft locomotion robots. *Journal of Mechanical Design*, 141(10), 2019.
- 451 [48] Tsun-Hsuan Wang, Pingchuan Ma, Andrew Everett Spielberg, Zhou Xian, Hao Zhang, Joshua B.
 452 Tenenbaum, Daniela Rus, and Chuang Gan. Softzoo: A soft robot co-design benchmark for
 453 locomotion in diverse environments. In *The Eleventh International Conference on Learning
 454 Representations*, 2023.
- 455 [49] Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi S Jaakkola. Crystal
 456 diffusion variational autoencoder for periodic material generation. In *International Conference
 457 on Learning Representations*, 2021.
- 458 [50] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, , and Jian Tang. GeoDiff: A
 459 geometric diffusion model for molecular conformation generation. In *International Conference
 460 on Learning Representations*, 2021.

- 461 [51] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided
462 human motion diffusion model. *arXiv preprint arXiv:2212.02500*, 2022.
- 463 [52] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and
464 Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint*
465 *arXiv:2210.06978*, 2022.
- 466 [53] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel
467 diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,
468 pages 5826–5835, 2021.