PNAS RESEARCH ARTICLE | PHYSICS OPEN ACCESS

Programming patchy particles for materials assembly design

Ella M. King[a,1], Chrisy Xiyu Du[b,c,1,2], Qian-Ze Zhu[b], Samuel S. Schoenholz[d,e], and Michael P. Brenner[b,d,3]

Edited by Herbert Levine, Northeastern University, Boston, MA; received August 14, 2023; accepted November 21, 2023

Direct design of complex functional materials would revolutionize technologies ranging from printable organs to novel clean energy devices. However, even incremental steps toward designing functional materials have proven challenging. If the material is constructed from highly complex components, the design space of materials properties rapidly becomes too computationally expensive to search. On the other hand, very simple components such as uniform spherical particles are not powerful enough to capture rich functional behavior. Here, we introduce a differentiable materials design model with components that are simple enough to design yet powerful enough to capture complex materials properties: rigid bodies composed of spherical particles with directional interactions (patchy particles). We showcase the method with self-assembly designs ranging from open lattices to self-limiting clusters, all of which are notoriously challenging design goals to achieve using purely isotropic particles. By directly optimizing over the location and interaction of the patches on patchy particles using gradient descent, we dramatically reduce the computation time for finding the optimal building blocks.

programmable assembly | automatic differentiation | self-assembly

Significance

The development of new materials has been a transformative force in shaping the modern world. The traditional approach to creating new functional materials relies on a combination of hard-won intuition and arduous labor to sift through the innumerable possibilities. However, as the functions we need grow increasingly complex, finding that rare substance amid all possible materials becomes increasingly difficult. Instead, we will need methods for the direct design of novel materials. Existing methods for inverse materials design are limited to extremely simple components. Here, we dramatically increase the complexity of the designable components, broadening the scope of materials we can design.

Significant efforts have been made toward designing synthetic materials that rival the complexity we observe in biological systems (1–7). However, many of the synthetic systems studied suffer from one of two fatal flaws: Either the system is too simple to be able to replicate complex behaviors or the system is too complex to be easily designable. By combining the principles that enable machine learning methods to efficiently navigate large parameter spaces with physics- and materials science–informed models, we introduce a system that both complex enough to capture desirable functional behavior and is amenable to inverse design.

One major line of inquiry toward designing complex functional materials focuses on materials with uniform spherical particles as components. While this approach has led to promising advances (8–12), the absence of directional interactions significantly limits the design space. In materials with spherical components, designed interactions either are too complicated to be experimentally realizable (8) or necessitate that every particle interact uniquely with every other particle in the system, which cannot be physically instantiated at scale (13). Conversely, a rich literature of work (13–16) has been conducted by running forward simulations of systems with highly complex components, such as proteins. The complexity of the components means individual simulations are extremely computationally intensive, making inverse design approaches untenable. Moreover, the design space for these systems is too vast to search effectively.

Breaking rotational symmetry of the component particles vastly increases the potential for materials' designability without relying on having a large number of particle types. Extensive research has been done on anisotropic particles, ranging from mapping out phase diagrams for hard particles with nontrivial shapes (17) to designing patchy particles to self-assemble open lattices such as cubic diamond structures (18, 19). The design space for anisotropic particles is immense (20), making brute force approaches to searching for desirable properties unsustainable for modern materials design. However, the dramatic advances brought about by the machine learning community have recently made it possible to search vast design landscapes for regions with desirable behavior.

We introduce an inverse design method for anisotropic particles that is both physics-based and fully differentiable. Our work has parallels to previous work in that we design interactions that lead to target self-assembled structures (8–12, 21). However, many of these approaches lead to highly complex interactions (8, 13, 21) which are inaccessible experimentally. In contrast, we move the complexity from the interactions to the component particle geometries, which may lead to more experimentally realizable designs. There have been encouraging initial efforts to inverse-design materials with anisotropic components, but existing methods are either highly system-dependent (22, 23), require

Author contributions: E.M.K., C.X.D., and M.P.B. designed research; E.M.K., C.X.D., Q.Z.-Z., and S.S.S. performed research; E.M.K., C.X.D., Q.Z.-Z., and S.S.S. contributed new reagents/analytic tools; M.P.B. supervised research; E.M.K., C.X.D., and Q.Z.-Z. analyzed data; and E.M.K., C.X.D., and M.P.B. wrote the paper.

Competing interest statement: M.P.B. is an employee Google Research and S.S.S. an employee of OpenAI.

This article is a PNAS Direct Submission.

[1]C.X.D. and E.M.K. contributed equally to this work.

[2]Present address: Department of Mechanical Engineering, University of Hawai'i at Mānoa, Honolulu, HI 96822.

[3]To whom correspondence may be addressed. Email: brenner@seas.harvard.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2311891121/-/DCSupplemental.

Published June 24, 2024.

large training datasets for each system of interest (24, 25), or necessitate that the complex system can be captured by a much simpler, lower-dimensional representation (26, 27). Our method is system independent, does not rely on large amounts of training data, and can capture complex systems that do not exhibit lower dimensional representations.

Here, we introduce a framework that capitalizes on the advances wrought by machine learning to broadly enable inverse design of complex, anisotropic systems. This framework end-to end differentiable and is system-independent within JAX-MD (28), a Molecular Dynamics (MD) engine with automatic differentiation (AD) (29) enabled. AD is the workhorse underlying the explosion in productivity in the machine learning community in recent decades. We introduce the ability to directly optimize over particle geometry and anisotropic interactions. We demonstrate the method specifically on patchy particles, a model system that is simple enough to design, yet rich enough to capture features such as directional interactions. In this paper, we first discuss the implementation of the method, and then, we demonstrate the versatility of the platform by showcasing three examples: i) stabilization of a Kagome lattice, ii) self-limiting 2D ring assembly, and iii) stabilization of 3D finite clusters. We include a python notebook (30) that demonstrates how to perform the optimization for the 3D finite cluster case. Direct gradient descent of building block properties will enable researchers to efficiently design novel materials with targeted properties and functions.

Method

MD simulations are a powerful tool for understanding micro- and nanoscale systems. When combined with inverse design methods, MD simulations can be used to design highly complex structures and materials properties, ranging from intricate crystal structures (11, 22, 31), finite clusters (24), phase transitions (32), and kinetics (33) in self-assembled systems.

Existing methods that combine MD with automatic differentiation are limited to simulations of isotropic particles (33), which significantly limits the design space of complex materials functions. However, many standard MD libraries, such as Highly Optimized Object-oriented Many-particle Dynamics (HOOMD)-blue (34) and Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) (35), offer support for simulations of nonisotropic objects called rigid bodies. Rigid bodies in MD are generally defined as a system of spheres with no internal degrees of freedom, which can be used to simulate building blocks that have arbitrary shape and directional interactions.

Here, we build on an existing software package, JAX-MD (28), that enables fully differentiable MD simulations. The original release of JAX-MD did not support building blocks or integrators with rotational degrees of freedom. Here, we enable both simulation and differentiation of systems with anisotropic particles.

To simulate and optimize over anisotropic particles, we extended all the available integrators in JAX-MD to account for rotational degrees of freedom, following the algorithm introduced in ref. 36. The technical details of threading the gradients through simulations of anisotropic particles, in addition to a detailed discussion of other features of the implementation, can be found in *SI Appendix*.

Our prototypical optimization procedure for inverse design with MD begins with specifying all the variables needed for a standard forward simulation. These variables include the system information, such as the number of particles, pair potential, and box size, as well as the integrator information, such as the integrator type, temperature, and step size. While any of

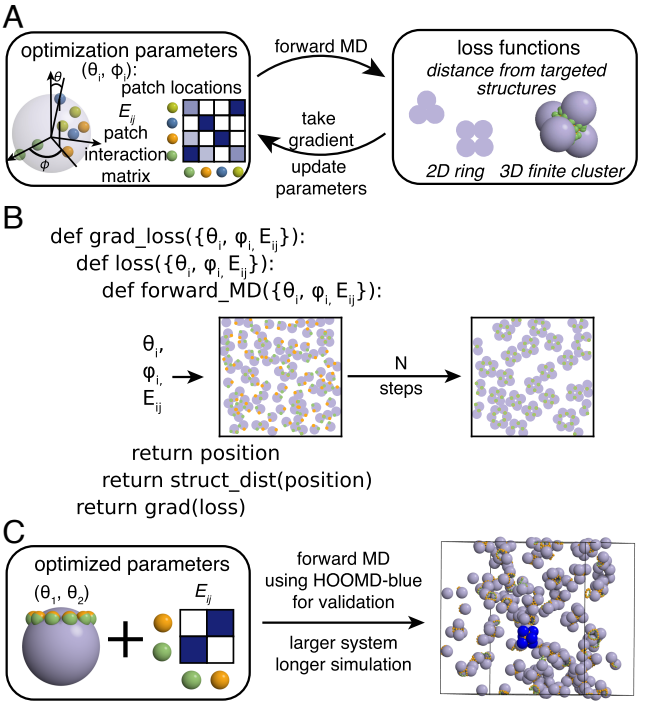

Fig. 1. (*A*) Optimizing patchy particle interactions. The optimization parameters (shown on the *Top*) are the patch locations and the interaction matrix of patch strengths. These parameters are used to run a forward simulation, and a loss function is computed. We then take the gradient of the loss function with respect to the optimization parameters and update the parameters accordingly. The loss functions vary for different optimization targets. (*B*) Gradient of loss function respect to parameters for optimization. The pseudo-code demonstrates how the gradient is computed based on the parameters for optimization. (*C*) Extrapolation to more performant MD engines. We test optimal parameters in HOOMD-blue, showing both that optimal parameters are valid across different MD engines and enabling rapid testing for longer simulations with more particles.

these variables can be optimized, we focus on examples where we optimize over only the pair potentials and the particle geometries. We parameterize the pair potential by a matrix of interaction strengths, and we parameterize the particle geometries by the locations of patches on central particles (Fig. 1*A*).

To compute gradients of patch locations (θ_i, ϕ_i) and interaction strengths (E_{ij}), we first define a forward MD simulation function (Fig. 1*B*) that takes $(\theta_i, \phi_i, E_{ij})$ as input parameters and returns the position data of every particle in the simulation. We then use the position data as input for our loss function, where we define criteria to evaluate whether the system is close to our targeted behavior. Finally, we take the gradient of the loss function with respect to $(\theta_i, \phi_i, E_{ij})$, and update $(\theta_i, \phi_i, E_{ij})$ based on the gradient values for the next iteration. We use the Adam optimizer to update our parameters based on the gradient computation.

As a final step, shown in Fig. 1*C*, we use our optimal parameters to run forward simulations in a more performant MD engine. These forward simulations are run for longer timescales and with more particles, demonstrating validity of our optimal parameters across different MD engines and beyond the time- and length-scale of the simulations we optimize over. This iterative procedure shows one possible means of bringing inverse-design into traditional MD simulation.

We note that this optimization procedure does not rely on any black-box methods: All the functions we differentiate are physics-based simulations. Because every step of the optimization function, from MD simulation to loss function evaluation, is

fully differentiable, we are able to compute gradients of arbitrary parameters.

Results

We demonstrate the optimization and design of patchy particles. These particles consist of a central particle plus a set of patches rigidly attached to the central particle (Fig. 1*A*). The central particle describes the general shape of the patchy particle, and the set of patches governs the directional interaction of the patchy particle. Because we have enabled end-to-end differentiable simulations of anisotropic particles, we can directly optimize over the locations of the patches and the interaction matrices between patches. As a result, the rich design space available to patchy particles becomes feasible to search.

Here, we showcase three examples: i) stabilization of a Kagome lattice, ii) self-limiting 2D ring assembly, and iii) stabilization of 3D finite clusters. Designing parameters for any of these three examples using isotropic particles is highly challenging and requires complex pair potentials. By introducing anisotropy, we demonstrate robust design of each of these systems with simple interaction potentials.

In each example discussed, the patches interact via a Morse potential, and the central particles interact via a soft sphere potential or Weeks-Chandler-Andersen (WCA) potential.

The three potentials are as follows:

Morse Potential:

$$U(r) = \varepsilon(1 - e^{-\alpha(r-r_0)})^2, \quad [1]$$

Soft Sphere Potential:

$$U(r) = \varepsilon\left(\frac{\sigma}{r}\right)^{\alpha}, \quad [2]$$

WCA Potential:

$$U(r) = \varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] + \varepsilon. \quad [3]$$

These interaction potentials are convenient for our use case but can be readily changed for different applications. In each optimization example, we specify a loss function describing our targeted materials properties. This loss function must itself be fully differentiable and thus cannot rely on discrete calculations.

Stabilizing a Kagome Lattice. The Kagome lattice (37) is an open lattice structure (Figs. 1 and 2*B*) with a broad array of potential materials applications (38, 39). Self-assembling a Kagome lattice from isotropic potentials requires complicated potential landscapes with both attractive and repulsive wells (40), which are rarely possible to instantiate experimentally. One way to simplify the assembly is to introduce anisotropy. Inspired by the experimental realization of a Kagome lattice using Triblock Janus spheres (41), we propose a general patchy particle model to optimize patch locations that stabilize a Kagome lattice, using a simple Morse potential.

Each component in the model consists of a central particle and six patches that are rigidly attached to the central particle (Fig. 2*B*). The central particles interact with one another via a soft sphere potential, and the patches each interact via a Morse potential. We keep the Morse interactions fixed and optimize over the locations of the patches on the central particle.

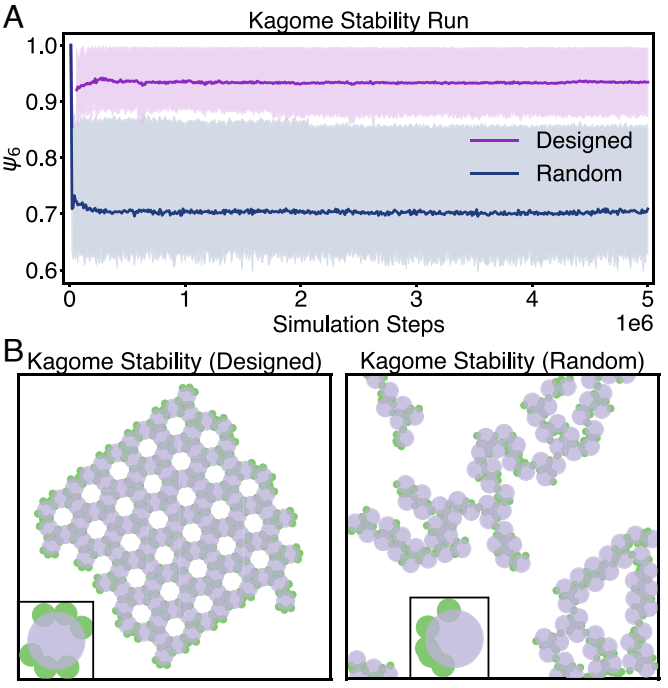

Fig. 2. Stabilizing a kagome lattice. (*A*) Stability of a system initialized in a kagome lattice configuration as a function of time. The purple (*Top*) results use our optimized parameters, while the navy (*Bottom*) results use random initial parameters. The optimization is initialized with random parameters. (*B*) Simulations run in HOOMD-blue demonstrating that the optimal parameters (*Left*) stabilize the kagome lattice, while random parameters (*Right*) cause the lattice to melt.

The optimization procedure follows the structure outlined in the Method section. We initialize the system in a Kagome lattice configuration with the orientations of the particles randomized. We then run 200 replicate MD simulations. The simulations are run with for 40,000 steps with a timestep (dt) of 1e−3 and 100 particles, at a temperature (kT) of 0.1 and an area fraction of 0.3 with periodic boundary conditions. We then measure the average loss function across the replicates. Because we initialize in the lattice configuration, the loss function for the optimization is simply the distance of the particles from their initial position: If the Kagome lattice is stable, the lattice will not melt, and the positions of the particles will remain constant, up to vibrational motion. We compute the gradient of the average loss function with respect to the positions of the patches on the central particle. Finally, we update the positions of the patches based on the value of the gradient using the Adam optimizer available in Autograd and XLA (Accelerated Linear Algebra) (JAX). We repeat this procedure for 100 optimization steps with a learning rate of 0.1, and then starting from the optimal value, we run another 100 optimization steps with a learning rate of 0.05 and finally another 100 steps with a learning rate of 0.01.

At the end of the JAX-based optimization procedure, we take the optimal parameters to run a longer simulation testing their stability using HOOMD-blue (34, 42, 43). Fig. 2*A* shows the ψ_6 order parameter measured using Freud (44) for designed building blocks and randomly generated ones. We can see clearly that the designed building blocks stabilize the Kagome lattice and the parameters are transferable across different MD engines.

Self-Limiting Rings. Despite the fact that natural systems rely on self-limited assembly, synthetically developing self-limiting structures remains a significant challenge (5). Using our model,

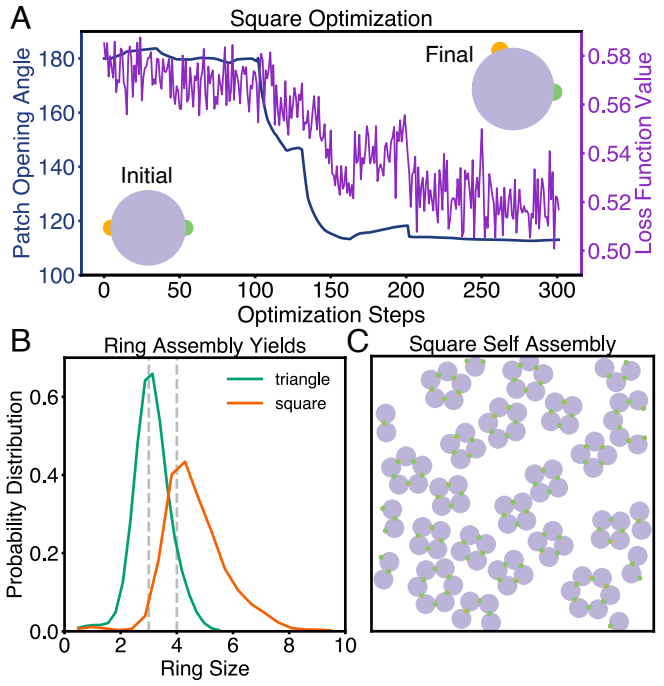

Fig. 3. Assembly of self-limiting rings. (*A*) Optimization results for formation of square rings. The *x*-axis shows the optimization steps. The *y*-axis shows the patch opening angle on the *Left* (navy), and the loss function on the *Right* (purple). Over the course of the optimization, the patch opening angle tends toward a value slightly greater than 100°. (*B*) Assembly yields computed from forward MD simulations run in HOOMD-blue for the patchy particles designed to yield triangles (green) and squares (orange). The yield of each ring design peaks at the desired ring size, demonstrating that the design procedure was successful. (*C*) One example end result of assembling squares in a bath of particles. While some incorrect products (primarily larger rings) are observed, most of the particles are in square configurations.

we design self-limited rings of varying sizes that self-assemble in a bath of components.

To design self-limiting rings, we use a patchy particle model that consists of a central particle and two patches. The central particles interact via a soft sphere potential Eq. **2**, and the patches interact via a Morse potential Eq. **1**. We optimize over both the location of the patches and the strength of the patch interactions. Each patch is allowed to vary independently. We initialize the patch positions and strengths randomly.

The optimization procedure again follows the structure listed in *Method*. Here, however, because we are interested in self-assembly rather than stabilization, we initialize the simulation with particles with random initial positions and orientations.

To compute the loss for this calculation, we take the distance between each particle and its M nearest neighbors ($M = 3$ for square rings because squares consist of four particles, etc), and compare those distances to a reference structure. The reference structure is a perfectly assembled ring. We optimize over both the positions and the strengths of interactions of the patches. We compute the average loss over 128 replicate simulations that each run for 40,000 steps with a dt of 1e−3, at a temperature (kT) of 1.0 and an area fraction of 0.2. To reduce computational cost, we optimize over only the last 1,000 steps of the simulation. Despite this approximation, the gradients are meaningful enough to converge to optimal parameters.

Our model rapidly converges to a set of parameters that consistently forms independent rings of the specified size (Fig. 3*A*). Evolution of interactions between patches during optimization is included in *SI Appendix*. While we do observe occasional malformed structures, we do not observe the formation of any extended structures in our system. We have thus successfully captured self-limiting behavior with differentiable patchy particles.

While one would naively assume that the optimal patch opening angle to form 4-component square rings would be 90°, we find that the optimal opening angle is significantly wider, as shown in Fig. 3. Our optimal results demonstrate a higher yield of square rings than the naive guess, as can be seen in Fig. 3, as well as in *SI Appendix*.

We hypothesized that the yield of squares is higher for a wider opening angle because it prevents the formation of triangles. To test this hypothesis, we performed two measurements. First, we measured the yield of triangles, squares, and pentagons in the system of particles designed to form squares. The results are given in Fig. 3 and in *SI Appendix*. We indeed observe that the formation of triangles is significantly suppressed for the designed parameters relative to the naive 90-degree guess.

Self-Limiting 3D Clusters. While our 2D examples were successful, working with three-dimensional structures often poses different challenges. Inspired by virus shells, we demonstrate the stabilization of the simplest nontrivial platonic solid: the octahedron (Fig. 4 *A*, *ii*). We leverage the nonisotropic interactions offered by patchy particles to find the patch positions and interaction strengths that consistently stabilize octahedral structures.

We begin with the model proposed by Long and Ferguson (24), consisting of two layers of patches in concentric circles (Fig. 4 *A*, *i*). We optimize over the positions of these circles of patches while fixing the interaction strength to be consistent with ref. 24. Critically, though ref. 24 required mapping out regions of the free energy landscape to achieve assembly of clusters, we are able to recover features similar to their results with no explicit measurement of the free energies.

Based on the simulation details in ref. 24, we adapted the model to the JAX-MD simulation environment. We use a WCA potential Eq. **3** for the center particle with $\sigma = 5.0$ and $\epsilon = 1.0$ and a Morse potential for the patches with $\epsilon = 4.0$ and $r_0 = 0.0$. We simulate using a Langevin integrator with $\gamma = 5.0$, $dt = 1e-4$, $kT = 0.8$, and a number density of 0.05. Despite the modifications to the simulation parameters, the energy scale and dynamics of our system closely follow those described in ref. 24.

Our optimization procedure for stabilizing 3D clusters closely mirrors our method in two dimensions, with minor modifications. We note that the self-assembly process for finite 3D clusters is considerably more complex from both thermodynamic and simulation perspectives. There are far more competing structures, and the system takes much longer to equilibrate. We mitigate these challenges by initializing our systems with six patchy particles, each consisting of one center particle and 20 patches, in a perfect octahedron.

We again use a loss function that consists of computing nearest neighbor distances relative to those of a reference structure. In this case, the reference structure is the correctly formed octahedron. For every optimization step, we run one simulation for 200,000 steps and compute the gradient of the loss function to update patch locations. The length of the simulation is determined by analyzing the time needed for to self-assemble a partial cluster (see *SI Appendix* for details) and the number of replicates per optimization step is decided by gradient magnitude. We unexpectedly observed that using Langevin dynamics over

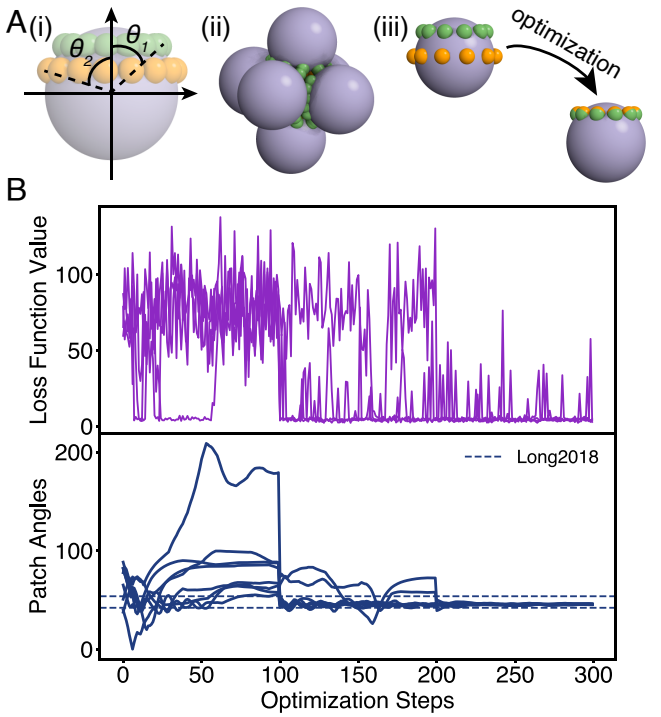

Fig. 4. Stabilization of octahedral cluster. (*A*) (*i*) patchy particle model used to optimize for octahedral cluster. θ_1 and θ_2 determine the locations of the two rings of patches, where each ring has A-A type attractions. (*ii*) sample octahedral cluster (*iii*) patchy particle evolution from the beginning to the end of an optimization run. (*B*) Ensemble of optimization results for stabilization of an octahedral cluster. The *x*-axis shows the optimization steps. The *y*-axis shows the patch opening angle on the *Bottom* plot (navy), and the loss function on the *Top* plot (purple). The dotted lines show the optimized parameters from ref. 24. Over the course of the optimization, the two patch angles converge to the same value and fall into the range of the literature optimized values. The set of independent optimization runs all converge to the same optimal patch angles. 3D particles and trajectories are rendered by INJAVIS (48).

Nosé-Hoover significantly reduces variation in the gradients. The whole optimization procedure consists of 300 optimization steps with three learning rates $[0.1, 0.05, 0.01]$ respectively using the Adam optimizer.

We initialize the optimization with randomly generated patch angle parameters (see Fig. 4 *A*, *iii* as an example). With these random parameter values, the octahedron is not stable. This can be seen in the early values of the loss function: The loss at the outset of the optimization is both large and highly variable. As the optimization proceeds, the patch positions converge and the loss decreases. Ultimately, the optimization converges to parameters that reliably stabilize the three-dimensional cluster, as shown in Fig 4*B*. We performed four independent optimizations for cluster stabilization, and for all the runs where the loss function converged, the patch locations converged to the same value also (see *SI Appendix* for details).

Because we optimize for stabilization rather than assembly, our results deviate slightly from ref. 24. In ref. 24, the optimal parameters for octahedra assembly is $[42.0°, 53.7°]$, while our optimal parameters for octahedra stabilization is $[45.3°, 46.0°]$ (see ref. 51 for simulation data). We conclude that having two rings at similar locations is more favorable than having separated rings for the case of stabilization and validate this conclusion with forward simulations of systems with each set of parameters. Indeed, our optimal parameters yielded a lower loss for stabilization than those in ref. 24 (see *SI Appendix* for details). This may be alternatively explained by a difference in the two models: Our patches have no volume of their own.

Despite these differences, our optimal ring positions fall between the two found in ref. 24. The optimal location we find is closer to the inner ring in ref. 24, which may indicate that the inner ring is a more vital feature in stabilization. Additionally, we ran forward self-assembly simulations using the two sets of parameters using HOOMD-blue to test their ability to self-assemble, and the JAX-MD optimized parameters lead to a faster decrease of our loss function (see *SI Appendix* for more details).

Critically, we were able to achieve these results without mapping the free energy landscape. This method not only provides a straightforward way to search the design space of anisotropic particles for properties of interest but also showcase how a small difference in model choice could lead to different optimal final results. This fast feedback loop for particle design could be instrumental in mapping to experimental systems.

Discussion

We have introduced an end-to-end differentiable model system capable of capturing rich functional behavior in materials while still being simple enough to directly design. We have demonstrated the model by designing stabilization of an open lattice structure, self-limiting assembly in 2D, and stabilization of 3D finite clusters.

In each case, we have made use of only one particle type. Previous efforts to, e.g., stabilize or assemble octahedral structures have relied on having N different particle types to assemble a structure of N particles (33, 45). Though our individual components are more complex, the need to construct only one particle type renders our model system possible to manufacture at scale.

Though we believe our model offers significant potential for design of novel functional materials, its design potential is limited by requiring differentiable loss functions and computational expense. To compute meaningful gradients based on the loss function, we can only use loss functions that do not rely on a sharp radial or nearest-neighbor cutoff. This limitation makes using traditional well-performed loss functions (order parameters), such as the local bond order parameter (46), less feasible and increases the difficulty of designing self-limiting structures, where one of the most straightforward loss functions is to count the number of particles in a cluster. The limitation on loss functions is both a challenge and an opportunity. With more machine learning techniques being incorporated in materials design, we need to rethink how we describe materials structures and properties and come up with new robust and accurate descriptors that fit the inverse-design method at hand.

On the side of computational expense, our method is limited mostly by Graphics Processing Unit (GPU) memory. Large systems of particles yield highly memory-intensive gradient computations, so smaller systems or behaviors that are well described by local structure are better suited to the model. The memory usage for a particular optimization run is decided by the combination of system size (N), run timesteps (r_s), and batch size (b). For a GPU with 32 Gb of memory, we can run a simulation of $N \times r_s \times b \leq 10^8$. The limitation can be mitigated in a few ways: run on a GPU with higher memory, distribute batch sizes on multiple GPUs, and distribute a single big system on multiple GPUs. Currently, there are GPUs that have higher

memory capacity, and JAX-MD already allows distributing batch sizes on multiple GPUs. We are actively working on parallelizing a single system on multiple GPUs to reduce the memory limitation.

We have only begun to explore the range of behaviors available to this model. One possible extension of our work on self-limited structures is to learn the rules of assembly for larger, multi-component virus-like particles, biomimetic shells that have the potential to be used for drug delivery. Additionally, while we made use of zero-width patches, if we instead considered patches that were physical particles, we could explore the realm of colloidal molecules. These structures have been instantiated experimentally, but the design space of colloidal molecules is vast and underexplored (6, 20, 47).

This model is the right paradigm to reach the longstanding goal of directly designing for functional behavior by optimizing small, simple components.

Data, Materials, and Software Availability. Simulation output data have been deposited in https://github.com/brennergroup/patchy_particle_data (51) (https://doi.org/10.5281/zenodo.10051842) (52). All other data are included in the manuscript and/or *SI Appendix*.

ACKNOWLEDGMENTS. We thank Carl Goodrich, Ryan Krueger, and Megan Engel for helpful discussions. This material is based upon work supported by the Office of Naval Research (ONR N00014-17-1-3029), the NSF Graduate Research Fellowship under Grant No. DGE1745303, NSF Grant DMR-1921619, and the NSF AI Institute in Dynamic Systems (#2112085) 3D particles and trajectories are rendered by INteractive JAva VISualization (INJAVIS) Engel (2021). Data generated in this paper are managed by Adorf et al. (2018) and Ramasubramani et al. (2018).

Author affiliations: [a]Department of Physics, Harvard University, Cambridge, MA 02139; [b]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02139; [c]Mechanical Engineering, University of Hawai'i at Mānoa, Honolulu, HI 96822; [d]Google Research, Mountainview, CA 94043; and [e]OpenAI, San Francisco, CA 94110

1. M. Sindoro, N. Yanai, A. Y. Jee, S. Granick, Colloidal-sized metal-organic frameworks: Synthesis and applications. *Accou. Chem. Res.* **47**, 459–469 (2014).
2. M. R. Jones, N. C. Seeman, C. A. Mirkin, Programmable materials and the nature of the DNA bond. *Science* **347** (2015).
3. W. B. Rogers, W. M. Shih, V. N. Manoharan, Using DNA to program the self-assembly of colloidal nanoparticles and microparticles. *Nat. Rev. Mater.* **1**, 1–14 (2016).
4. Z. Zeravcic, V. N. Manoharan, M. P. Brenner, Colloquium: Toward living matter with colloidal particles. *Rev. Mod. Phys.* **89**, 031001 (2017).
5. M. F. Hagan, G. M. Grason, Equilibrium mechanisms of self-limiting assembly. *Rev. Mod. Phys.* **93**, 025008 (2021).
6. T. Hueckel, G. M. Hocky, S. Sacanna, Total synthesis of colloidal matter. *Nat. Rev. Mater.* **6**, 1053–1069 (2021).
7. C. X. Du *et al.*, Programming interactions in magnetic handshake materials. *Soft Matter* **18**, 6404–6410 (2022).
8. J. Dshemuchadse, P. F. Damasceno, C. L. Phillips, M. Engel, S. C. Glotzer, Moving beyond the constraints of chemistry via crystal structure discovery with isotropic multiwell pair potentials. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2024034118 (2021).
9. H. Fang, M. F. Hagan, W. B. Rogers, Two-step crystallization and solid-solid transitions in binary colloidal mixtures. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 27927–27933 (2020).
10. W. D. Piñeros, B. A. Lindquist, R. B. Jadrich, T. M. Truskett, Inverse design of multicomponent assemblies. *J. Chem. Phys.* **148**, 104509 (2018).
11. B. A. Lindquist, R. B. Jadrich, W. D. Piñeros, T. M. Truskett, Inverse design of self-assembling frank-kasper phases and insights into emergent quasicrystals. *J. Phys. Chem. B* **122**, 5547–5556 (2018).
12. C. S. Adorf, J. Antonaglia, J. Dshemuchadse, S. C. Glotzer, Inverse design of simple pair potentials for the self-assembly of complex structures. *J. Chem. Phys.* **149**, 204102 (2018).
13. S. Angioletti-Uberti, B. M. Mognetti, D. Frenkel, Theory and simulation of DNA-coated colloids: A guide for rational design. *Phys. Chem. Chem. Phys.* **18**, 6373–6393 (2016).
14. T. E. Ouldridge, A. A. Louis, J. P. Doye, Structural, mechanical, and thermodynamic properties of a coarse-grained DNA model. *J. Chem. Phys.* **134**, 02B627 (2011).
15. K. J. Bowers *et al.*, "Scalable algorithms for molecular dynamics simulations on commodity clusters" in *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing* (2006), p. 84-es.
16. J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, D. A. Case, Development and testing of a general amber force field. *J. Comput. Chem.* **25**, 1157–1174 (2004).
17. D. Klotsa, E. R. Chen, M. Engel, S. C. Glotzer, Intermediate crystalline structures of colloids in shape space. *Soft Matter* **14**, 8692–8697 (2018).
18. F. Romano, J. Russo, L. Kroc, P. Šulc, Designing patchy interactions to self-assemble arbitrary structures. *Phys. Rev. Lett.* **125**, 118003 (2020).
19. A. B. Rao *et al.*, Leveraging hierarchical self-assembly pathways for realizing colloidal photonic crystals. *ACS Nano* **14**, 5348–5359 (2020).
20. S. C. Glotzer, M. J. Solomon, Anisotropy of building blocks and their assembly into complex structures. *Nat. Mater.* **6**, 557–562 (2007).
21. S. Whitelam, I. Tamblyn, Neuroevolutionary learning of particles and protocols for self-assembly. *Phys. Rev. Lett.* **127**, 018003 (2021).
22. Y. Geng, G. van Anders, P. M. Dodd, J. Dshemuchadse, S. C. Glotzer, Engineering entropy for the inverse design of colloidal crystals from hard shapes. *Sci. Adv.* **5**, eaaw0514 (2019).
23. X. Zhang, D. M. Tartakovsky, Optimal design of nanoporous materials for electrochemical devices. *Appl. Phys. Lett.* **110** (2017).
24. A. W. Long, A. L. Ferguson, Rational design of patchy colloids via landscape engineering. *Mol. Syst. Design Eng.* **3**, 49–65 (2018).
25. M. Röding, V. W. Skärström, N. Lorén, Inverse design of anisotropic spinodoid materials with prescribed diffusivity. *Sci. Rep.* **12**, 17413 (2022).
26. Y. Ma, A. L. Ferguson, Inverse design of self-assembling colloidal crystals with omnidirectional photonic bandgaps. *Soft Matter* **15**, 8808–8826 (2019).
27. D. E. Pinto, P. Šulc, F. Sciortino, J. Russo, Design strategies for the self-assembly of polyhedral shells. *Proc. Natl. Acad. Sci. U.S.A.* **120**, e2219458120 (2023).
28. S. Schoenholz, E. D. Cubuk, JAX MD: A framework for differentiable physics. *Adv. Neural Inf. Process. Syst.* **33**, 11428–11441 (2020).
29. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: A survey. *J. March. Learn. Res.* **18**, 1–43 (2018).
30. https://github.com/brennergroup/paper_notebooks/blob/main/Octahedron_Stabilization.ipynb (2023).
31. G. van Anders, D. Klotsa, A. S. Karas, P. M. Dodd, S. C. Glotzer, Digital alchemy for materials design: Colloids and beyond. *ACS Nano* **9**, 9542–9553 (2015).
32. C. X. Du, G. van Anders, J. Dshemuchadse, P. M. Dodd, S. C. Glotzer, Inverse design of compression-induced solid-solid transitions in colloids. *Mole. Simul.* **46**, 1037–1044 (2020).
33. C. P. Goodrich, E. M. King, S. S. Schoenholz, E. D. Cubuk, M. P. Brenner, Designing self-assembling kinetics with differentiable statistical physics models. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2024083118 (2021).
34. J. A. Anderson, J. Glaser, S. C. Glotzer, HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations. *Comput. Mater. Sci.* **173**, 109363 (2020).
35. A. P. Thompson *et al.*, LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.* **271**, 108171 (2022).
36. T. Miller Iii *et al.*, Symplectic quaternion scheme for biophysical molecular dynamics. *J. Chem. Phys.* **116**, 8649–8659 (2002).
37. I. Syôzi, Statistics of Kagomé lattice. *Progr. Theor. Phys.* **6**, 306–308 (1951).
38. T. H. Han *et al.*, Fractionalized excitations in the spin-liquid state of a Kagome-lattice antiferromagnet. *Nature* **492**, 406–410 (2012).
39. H. Xue, Y. Yang, F. Gao, Y. Chong, B. Zhang, Acoustic higher-order topological insulator on a Kagome lattice. *Nat. Mater.* **18**, 108–112 (2019).
40. W. D. Piñeros, M. Baldea, T. M. Truskett, Designing convex repulsive pair potentials that favor assembly of Kagome and snub square lattices. *J. Chem. Phys.* **145**, 054901 (2016).
41. Q. Chen, S. C. Bae, S. Granick, Directed self-assembly of a colloidal Kagome lattice. *Nature* **469**, 381–384 (2011).
42. T. D. Nguyen, C. L. Phillips, J. A. Anderson, S. C. Glotzer, Rigid body constraints realized in massively-parallel molecular dynamics on graphics processing units. *Comput. Phys. Commun.* **182**, 2307–2313 (2011).
43. J. Glaser, X. Zha, J. A. Anderson, S. C. Glotzer, A. Travesset, Pressure in rigid body molecular dynamics. *Comput. Mater. Sci.* **173**, 109430 (2020).
44. V. Ramasubramani *et al.*, freud: A software suite for high throughput analysis of particle simulation data. *Comput. Phys. Commun.* **254**, 107275 (2020).
45. Z. Zeravcic, V. N. Manoharan, M. P. Brenner, Size limits of self-assembled colloidal structures made using specific interactions. *Proc. Natl. Acad. Sci. U.S.A.* **111**, 15918–15923 (2014).
46. P. J. Steinhardt, D. R. Nelson, M. Ronchetti, Bond-orientational order in liquids and glasses. *Phys. Rev. B* **28**, 784 (1983).
47. A. Walther, A. H. Muller, Janus particles: Synthesis, self-assembly, physical properties, and applications. *Chem. Rev.* **113**, 5194–5261 (2013).
48. M. Engel, Injavis–interactive java visualization (2021). https://doi.org/10.5281/zenodo.4639570.
49. C. S. Adorf, P. M. Dodd, V. Ramasubramani, S. C. Glotzer, Simple data and workflow management with the signac framework. *Comput. Mater. Sci.* **146**, 220–229 (2018).
50. V. Ramasubramani, C. S. Adorf, P. M. Dodd, B. D. Dice, S. C. Glotzer, "signac: A Python framework for data and workflow management" in *Proceedings of the 17th Python in Science Conference* (2018), pp. 152–159.
51. E. M. King, C. X. Du, Q.-Z. Zhu, S. S. Schoenholz, M. P. Brenner, brenner-group/patchy_particle_data: Data for Programmable patchy particles for materials design. GitHub. https://github.com/brennergroup/patchy_particle_data. Deposited 29 October 2023.
52. C. X. Du, Data for Programmable patchy particles for materials design. Zenodo. https://doi.org/10.5281/zenodo.10051842. Deposited 29 October 2023.