

Patchy Particles Design: From Floppy Modes to Sloppy Dimensions

Gregory Snyder¹ and Chrisy Xiyu Du^{1,*}

¹*Mechanical Engineering, University of Hawai'i at Mānoa, Honolulu, HI, 96822*

(Dated: October 13, 2025)

Patchy particles have proven to be a prominent model for studying the self-assembly behavior of various systems, ranging from finite clusters to bulk crystal assemblies, and from synthetic colloidal particles to viruses. The patchy particle model is flexible, but it also comes with its own pitfalls—the potential design space is infinite. Many efforts have been put into building inverse-design frameworks that efficiently design patchy particles for targeted assembly behaviors. In contrast, little work has been done on investigating the interplay between different types of parameters that can be optimized for patchy particles, such as patch location, patch size, and patch binding energies. Here, by utilizing molecular dynamics with automatic differentiation, we elucidate the relationships between different potential optimization parameters and provide general guidelines on how to approach patchy particle design for various types of finite clusters. Specifically, we find that the design parameter landscape is highly dependent on the floppiness of the target structure, and we can identify stiff and sloppy parameters by computing the Hessians of all optimization parameters.

I. INTRODUCTION

Patchy particles [1], broadly defined as particles with discrete, attractive patches, are an excellent model system for studying the self-assembly behaviors of soft material systems, ranging from colloidal particles [2–5] to proteins [6–10]. Known for their versatility, there are many realizations of patchy particles: triblock spheres that assemble into Hexagonal or Kagome lattice depending on the patch size [11, 12], patchy rhombuses and hexagons with programmable edges that assemble into both finite and bulk structures [13, 14], spheres with binding sites that assemble into Diamond structure [15, 16], and triskelions that mimic the Clathrin protein assembly process [17]. Together with advances in theory and modeling of patchy particles, researchers have also made significant progress in patchy particle synthesis, such as functionalizing DNA patches on colloids [18] or vesicles [19], colloidal fusion [20], and shape-shifting patchy particles [21]. Other experimental materials platforms, such as magnetic handshake materials [22, 23], DNA origami triangles [24, 25], and even *de novo* proteins [26, 27], can be coarse-grained into patchy particle-like systems to understand their assembly behaviors.

On the other hand, the versatility of the patchy particle model comes with a vast design space. There are many parameters that one could consider varying in a patchy particle system: the number of patches, the location of patches, the interaction between patches, and the shape of the particle. While the number of patches is discrete, all other parameters can be varied continuously, thus creating a high-dimensional design space. To effectively search through the giant design space for a patchy particle configuration with a targeted materials property in mind, various inverse-design strategies have been developed. Covariance matrix adaptation evolu-

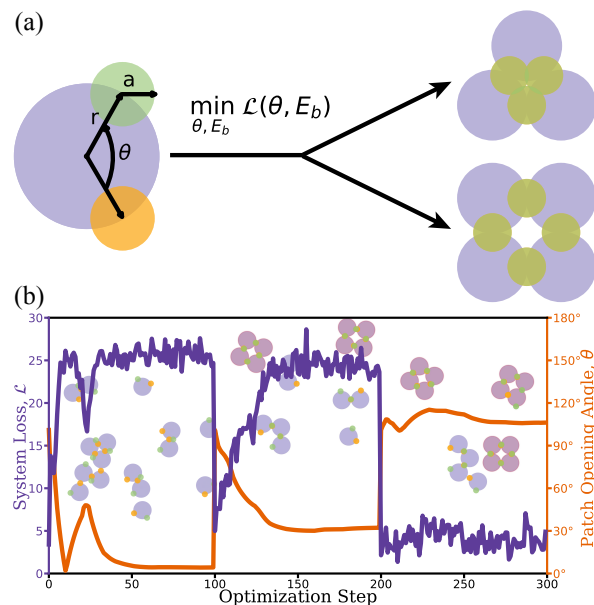


FIG. 1. System design (a) We use a particular patchy particle model with one big sphere (radius r) as the body and two patches (radius a) A (green) and B (orange) as the attractive binding sites. The locations of the two patches are determined by the opening angle θ , and the binding energies of the two patches are determined by the binding energy matrix E_b . We define a loss function $\mathcal{L}$ to find the best parameter combinations to assemble either a finite triangle or square robustly. (b) Parameter evolution for a typical self-assembly optimization with three different learning rates. At the beginning of every new learning rate, we select the best parameters from the previous learning rate as the starting point. The insert simulation snapshots show the final states of a sample batch. As the optimization converges, more squares start to form.

tionary strategy (CMA-ES) was employed to design the optimal patchy particle for finite cluster assembly [28] and to optimize photonic band gaps for colloidal crystals [29]. Neural Networks, combined with iterative learn-

* xiyudu@hawaii.edu

ing, are used to design various self-assembly motifs in 2D patchy particle systems [30]. Boolean satisfiability problem (SAT) solvers were used to design patchy particles to better assemble the Diamond structure [31, 32], and Digital Alchemy with incorporated patchy interaction [33] was able to find designs for the Kagome lattice.

While these methods have been effective, one component that is hard to account for is the self-assembly kinetics. Many inverse-design methods are very good at finding a design that minimizes the free energy of the system, but do not guarantee a pathway for the system to successfully reach the targeted structure [33, 34]. To overcome this, inverse-design strategies that combine automatic differentiation and Molecular Dynamics (MD) can implicitly or explicitly account for self-assembly kinetics or pathways. Using JAX-MD [35], an end-to-end differentiable MD engine, researchers were able to design colloidal cluster transition rates [36], configurations for fast disassembly [37], and pathways for work-minimization in non-equilibrium systems [38]. With the recent addition of rotational degrees of freedom in JAX-MD [39], it can now perform inverse-design for patchy particles for any arbitrary parameter combination.

In this paper, we demonstrate how to use automatic differentiation to systematically investigate the parameter design space for patchy particles, rather than just finding the “optimal” set of parameters. We employ a simple patchy particle model with two patches and optimize it for finite triangle and square assemblies. This approach enables us to uncover complex relationships among all potential design parameters, including patch size, patch opening angle, and patch binding energy. We provide detailed analysis to reveal the relationship between design parameters and physical properties (floppy vs. rigid) of the target structure. Lastly, we discuss a potential route to systematically examine the correlations between design parameters and identify stiff versus sloppy dimensions using Hessian information during the inverse design process.

II. METHOD

All the inverse-design and forward MD simulations in this paper are performed using JAX-MD [35]. JAX-MD is an MD engine that is end-to-end differentiable and GPU-accelerated. In particular, all our simulations utilize the `RigidBody` class developed in [39], allowing us not only to simulate anisotropic building blocks but also to directly take gradients on building block shapes. Fig. 2 shows a flowchart of how gradients are being threaded through from the beginning of the MD simulation to the end.

Our patchy particle model (see Fig. 1(a)) consists of three spheres of different species: a large center particle with two attractive patches (type A and B) along the surface. The locations of the two patches are determined by an opening angle θ . The center particle determines the

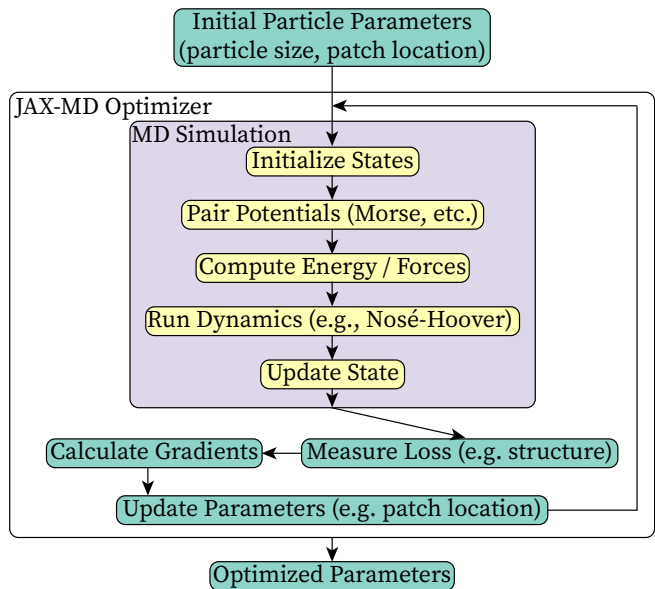


FIG. 2. **JAX-MD optimization framework:** the flowchart shows how the parameters for optimization are threaded through the simulation, allowing us to retrieve the gradients after a complete forward MD simulation.

shape and size of the patchy particle, which is modeled using the soft sphere potential [40]. We maintain a mass of 1.0, a radius of 1.0 ($\sigma = 2.0$), $\epsilon_s = 1.0$ and $\alpha_s = 2.0$ of the center particle for all our simulations. The two attractive patches are massless, and the size of the patch is decided by the α parameter in the Morse potential [41]. Soft sphere potential and Morse potential are defined as follows:

Soft Sphere Potential:

$$U_s(r) = \epsilon_s \left(\frac{\sigma}{r} \right)^{\alpha_s} \quad (1)$$

Morse Potential:

$$U_M(r) = \epsilon_m (1 - e^{-\alpha(r-r_0)})^2 \quad (2)$$

We explored ten different patch sizes initially for their assembly behaviors with $\alpha = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$. These α values correspond to patch size ratios $a/r = [0.88, 0.66, 0.53, 0.44, 0.38, 0.33, 0.29, 0.26, 0.24, 0.22]$. Detailed conversions between α and a/r can be found in the Supplementary Information (SI). Parameter ϵ_m governs the binding strength between all the patches, and we will use E_{AA} , E_{AB} , and E_{BB} to denote the interactions between patch types A and B.

In order to understand the relationship between patch size, opening angle, and binding energies for finite cluster assembly, we performed two different types of optimization runs: self-assembly and stability optimization. For self-assembly optimizations, we initialize the system with 15 or 18 particles (for triangles) or 16 particles (for squares). We initialize the system with all particles arranged randomly in a 2D box. For stability optimizations, we initialize the system with three particles (for

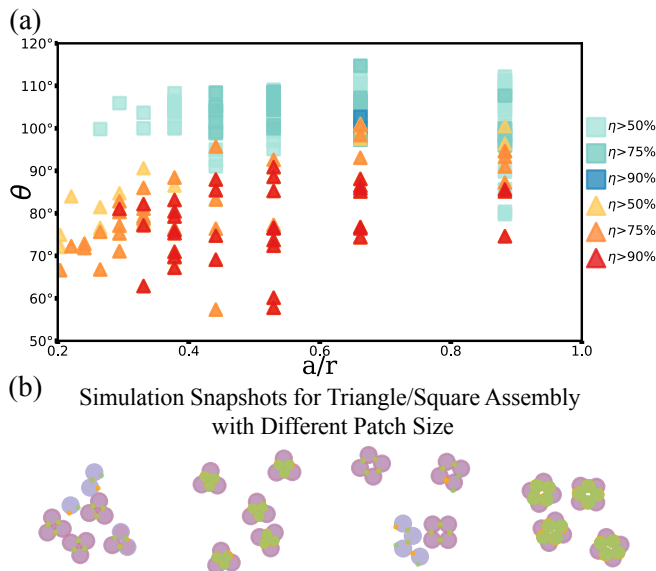


FIG. 3. **Self-Assembly Optimization Results:** (a) Optimal patch opening angle with respect to different patch sizes for triangle and square assembly. The square markers indicate results for square assembly, while the triangle markers indicate results for triangle assembly. The color of the markers corresponds to the yield η of a given optimization. We classify an optimization to be successful for $\eta > 50\%$ and we plot three η cutoffs: [50%, 75%, 90%]. (b) Four sample snapshots of the forward simulation for the last step of the optimization. We note successful triangle and square formations for both small and large patch sizes.

triangles) or four particles (for squares). We deliberately arrange the particles with desired positions and randomized orientations. For both types of optimizations, we run the optimization for 300 steps using the `adam` optimizer [42] with three different learning rates: 0.5, 0.1, and 0.05, respectively, for 100 steps each. When we switch to the new learning rate, we find the parameters from the last 100 steps that yielded the most correct structures and use those values as the starting point for the new learning rate. For each optimization step, we ran forward simulations with a batch size of 16 for stability optimization and a batch size of 64 for self-assembly optimization, both with 40,000 or 60,000 timesteps and a time step of $dT = 1 \times 10^{-3}$ at a density of $\rho = 0.2$ and a temperature of $kT = 1.0$. Fig. 1(b) shows the parameter and loss value evolution for a typical optimization run. The number of batches was determined by performing benchmark simulations that took into consideration reducing gradient noise and computational time (see details in SI).

Since the JAX-MD optimization process requires end-to-end differentiability, we cannot use a standard clustering algorithm as the loss function for the optimization. Instead, we match the local environment of each particle to a reference shape. Using square optimization as an example, we compute the three nearest neighbor distances of each particle in the final step of the simulation and

calculate the relative distance difference compared to a perfect square. The final loss value is the sum of all the differences between all particles in the simulation. If the simulation formed all squares, the system loss should be minimized to 0. However, in most self-assembly simulations, not all particles can successfully form into a square, resulting in a loss that is significantly greater than zero.

Additionally, given the definition of the loss function, a final configuration of all wrong structures might result in a lower loss than a configuration of one square and 12 monomers. To ensure that the final optimized parameters will favor square formation, we apply a secondary measurement after each optimization step: using the clustering algorithm in `freud` [43] to compute the number of squares formed in each batch. We then compute the yield of the optimization as the number of batches with at least one square formed, divided by the total number of batches.

III. RESULT

Self-Assembly Optimization: We first explored the effect of patch sizes on optimal patch opening angles for finite triangle and square self-assembly. In the previous study that introduced patchy particle optimization using JAX-MD [39], a single patch size was used during optimization. It was found that the particles preferred a patch opening angle bigger than the intuitive guess of 90° for squares and 60° for triangles. Here, we aim to investigate whether this result holds for other patch sizes or if there are other dependencies between the optimal patch size and opening angle.

Fig. 3(a) shows the relationship between patch opening angle and patch size ratios for both triangle and square assembly. The square markers show the optimization results for square assemblies of various patch sizes, while the triangle markers show the optimization results for triangle assemblies of various patch sizes. We note that there are different numbers of points for each patch size. The reason behind this is that we only plot the data that we label as successful here, and there are many optimization runs that either failed to converge or did not pass our success measurement bar. To measure success, we measure the yield (η) of triangles or squares during the optimization. η is computed as the number of batches that had at least one square/triangle formed in the last step of optimization over the total number of batches. We consider an optimization successful when η exceeds 50%. In Fig. 3(a), we separate the data points by three different η cut-offs: [50%, 75%, 90%] and color them accordingly. We note that as the patch size decreases, fewer optimizations were successful, which is due to kinetic effects during the simulation. As the patch size decreases, the chance of two patches being near each other also decreases, leading to a potentially longer binding time scale. To reduce kinetic effects, we increased the forward simulation time from 40,000 steps to 60,000 steps, allowing

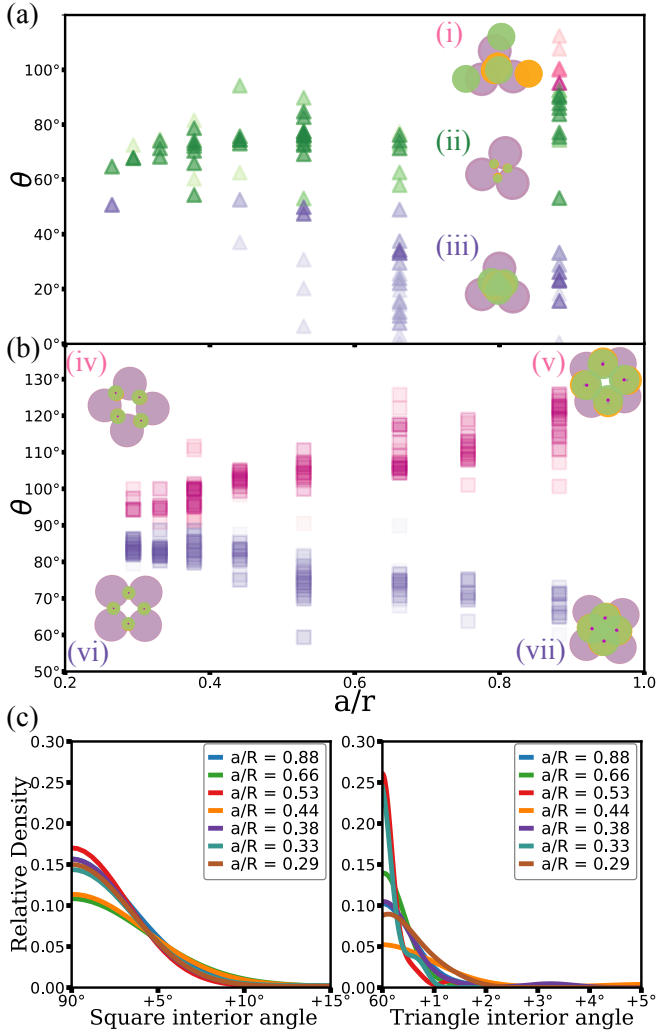


FIG. 4. **Stabilization Optimization Results:** (a) Optimal patch opening angle with respect to different patch sizes for triangle stabilization. Each data point represents one successful stabilization optimization. The darker regions indicate multiple optimizations converged to the same patch opening angle. The data points are categorized into three colors: (i) pink, (ii) green, and (iii) purple, indicating three different stable triangle morphologies. (b) Optimal patch opening angle with respect to different patch sizes for square stabilization. Each data point represents a successful stabilization optimization. The darker regions indicate multiple optimizations converged to the same patch opening angle. The data points are categorized into two types: (iv-v) pink and (vi-vii) purple, indicating the two different square conformations. (c) Interior angle distribution for squares and triangles. While the interior angle distribution is agnostic of patch sizes, the angle distribution for squares is dramatically bigger than that of triangles.

us to obtain more data points.

Looking at the relationship between patch opening angle and patch size, we notice two dramatically different trends for squares and triangles. As the patch size decreases, the optimal opening angle for triangles also de-

creases and approaches the intuitive guess of 60°. However, for square assemblies, we do not observe a similar trend. The optimal patch opening angles hover around 100°, and the only difference for different patch sizes is that the variance of optimal angles decreases as the patch size decreases. This result indicates that while optimizing for triangles and squares seems like the same type of inverse-design tasks, there is something fundamentally different between the two types of structures, causing different kinds of dependencies between the optimization parameters.

Lastly, we want to point out the ability of patch opening angles to select for different final assembled structures. Intuitively, as the patch size increases, the selective power of the patch opening angle will decrease, and no parameter combination can robustly assemble either squares or triangles. However, based on our results, we will lose the selective power only when the patch size is more than 60% of the size of the body particle. This result informs us that as long as the patch opening angle is controlled, we can allow for a bigger patch to accelerate the assembly process. Additionally, the patch opening angle overlapping region also validates the robustness of our inverse-design method, as we can isolate the structure we are optimizing for, even though the patchy particle can form different polymorphs.

Stabilization Optimization: Since the correlation of patch opening angle and patch sizes differs significantly between triangle and square assemblies. We believe that the difference arises from the structural property differences between triangles and squares. Using our patchy particle model, the assembled triangle clusters are *rigid*, while the assembled square clusters are *floppy* [44]. To confirm our hypothesis, we performed stabilization optimizations for various patch sizes to see if the same trends hold compared to the self-assembly optimizations.

Fig. 4(a) shows the optimal opening angle for triangle stabilization of different patch sizes. From first glance, there is almost no trend between patch size and opening angle, apart from the increasing range of possible opening angles as patch sizes increase. We went through all the optimization simulations and were able to identify three different triangle morphologies (see Fig. 4(i)-(iii)) that are equivalent using our position-based loss function. Morphology (i) only occurs at the larger patch sizes, as it uses only one of the two patches as binding sites for the triangle cluster. Morphology (ii) is the desired configuration to have AB binding, while all six patches bind with each other in morphology (iii).

Comparing the triangle stabilization results to the self-assembly ones, we notice that the trend of morphology (ii) generally agrees with the trend of triangle self-assembly. Morphology (i) is only a realistic configuration when no other particles exist in the system, as it is not self-limiting, and morphology (iii) is less kinetically favorable, as all patches need to lock into the same location. Combining the inverse-design results from both stabilization and self-assembly allows us to explore all the

possible configurations and separate energy and kinetics contributions to the final optimized parameters.

Fig. 4(b) shows the optimal opening angle for square stabilization of different patch sizes. Unlike the triangle results, square stabilization results show a different trend. As patch sizes increase, a clear bifurcation appears for the optimal patch opening angle. If we extrapolate the trends of the patch opening angle, it will converge to 90° as the patches reach a point particle. Fig. 4(iv)-(vii) show the configurations for square stabilization with high and low patch opening angles and small and big patch sizes. All four configurations fall under the same morphology with AB binding, the same as that of square self-assembly. We note that the stabilization results have a significant overlap with the self-assembly result for morphologies that appear in both sets of optimizations, and we provide a more detailed discussion in the SI.

The stabilization optimizations allow us to explore all possible morphologies for triangle and square cluster assembly. While the two structures are relatively simple, we were able to identify seven different configurations that are hard to pinpoint with traditional forward MD simulations. Comparing the morphologies between triangles and squares, we see that morphologies (i) and (iii) do not occur in square structures, as they require the structure to be space-filling. At the same time, we do not observe bifurcations in morphology (ii), as triangle structures are not floppy.

To confirm that square structures are indeed floppy while triangle structures are not, we measure the interior angle fluctuation of square and triangle structures (see Fig. 4(c)) for different patch sizes in equilibrium MD simulations. We see that *floppiness* is an inherent structural property, thus independent of the building block properties (patch size and opening angles), and the distribution of square interior angles is much broader than that of triangles.

IV. DISCUSSION

By systematically investigating how the optimal patch opening angle depends on patch sizes for different self-assembled finite structures, we discovered that the correlations between potential design parameters can be drastically different and highly depend on the structural properties of the final assembled structure. Moreover, we demonstrate that gradient-based optimization tools such as JAX-MD not only can help us inverse-design “optimal” building block parameters for a given assembly task, but they can also serve as a tool to aid the understanding of parameter landscapes in a high-dimensional design space.

In soft materials synthesis, oftentimes it is not possible to control all particle parameters precisely [45–47]. While many inverse-design methods provide a set of optimal parameters to achieve a specific material’s assembly or behavior [28, 31, 33, 34, 36, 39], it is also essential

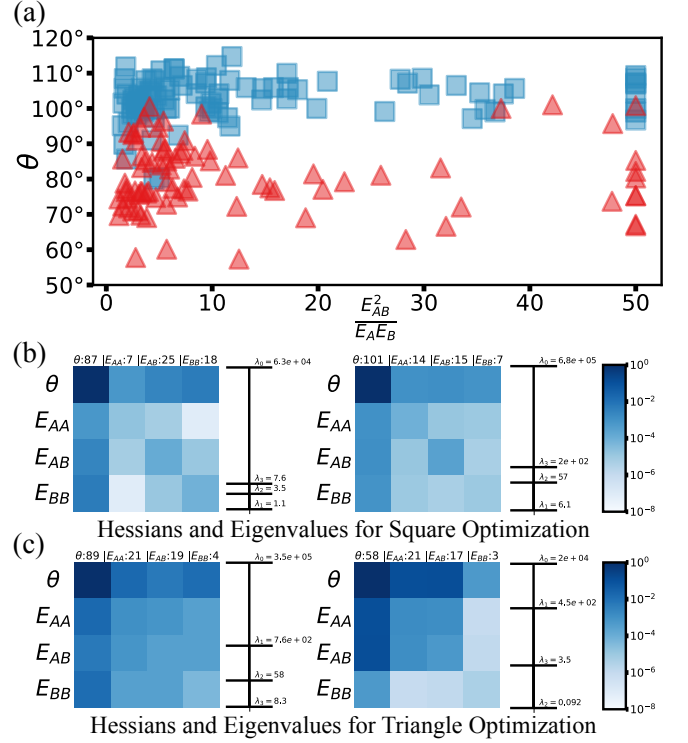


FIG. 5. **Relationship Between Optimized Patchy Particle Parameters:** (a) Scatter plot of optimized patch opening angle as a function of on-target binding and off-target binding energy ratio ($E_{AB}^2 / (E_{AA} E_{BB})$). Each data point represents one successful self-assembly optimization (square marker for square optimization and triangle marker for triangle optimization). The binding energy ratios are plotted with an upper limit of 50. (b-c) Hessians for the four optimized parameters (patch opening angle, E_{AA} , E_{AB} , and E_{BB}) and the corresponding eigenvalues for two different successful square and triangle optimizations. The optimized parameters are listed at the top of the Hessian matrices.

to discuss error tolerance. Our results not only discuss the possible single-parameter range that can lead to the same assembled structure, but also examine the coupled relationships between different parameters, such as patch size and patch opening angle. Our results can guide the synthesis of soft materials, whether it involves colloidal patchy particles [20], DNA-tethered particles [18], or *de novo* proteins [26]. For example, suppose one wishes to synthesize building blocks to self-assemble a rigid structure. In that case, it is crucial to have control over *both the size and location* of the binding sites, as demonstrated in our triangle optimization results: the larger the patch size, the larger the opening angle. On the other hand, if one wishes to synthesize building blocks for floppy structure assembly, then the binding site *location is the only important parameter*, and the size of the binding sites can be more forgiving. Having precise control over particles can be expensive, and our results can guide us on when low precision is good enough.

On the other hand, the parameters we chose to opti-

mize are driven by the physical properties of soft material systems, and we do not know the coupling among these parameters *a priori*. In previous work [39], it was demonstrated that the binding energies do not change significantly during optimization. In this work, our systematic investigations also result in the same conclusion (see details in SI). However, after plotting optimized patch opening angle as a function of final binding energy ratio between on-target and off-target binding (see Fig. 5(a)), we noticed that when the off-target binding energy (E_{AA} and E_{BB}) was strong, there is a wider spread in possible patch opening angles. While the on-target binding energy (E_{AB}) dominates, the possible patch opening angle narrows. These behaviors not only indicate a nontrivial dependence between binding energy strength and patch locations, but they also suggest that we may not need to eliminate off-target binding energies for successful assembly.

We further examine the coupling between binding energy and patch opening angle by computing the Hessian matrix of all possible mixed second derivatives of the loss function (see Fig. 5(b-c)) for triangle and square self-assembly optimization. We observed that the patch opening angle clearly plays a dominant role in the design optimization process, and the eigenvalues of the Hessians span several orders of magnitude. These behaviors are remarkably similar to the literature on information geometry, where the Hessian is equivalent to the Fisher information matrix in fitting a nonlinear model to data [48]. The large eigenvalues correspond to the *stiff* (important) parameter combinations that significantly affect the model prediction, while the small eigenvalues correspond to the *sloppy* (unimportant) parameter combinations underlying emergent simplicity of models [49]. The spread of eigenvalues across many orders of magnitude is a generic behavior of multiparameter models [50], corresponding to a valley-like loss landscape: descent across the valley slope is very fast, but further navigation along the valley length is slow.

The conceptual goal of design optimization differs somewhat from that of statistical inference: while inference can largely disregard specific parameter values in sloppy directions, in design, the sloppy parameters are where design freedom resides. To highlight the importance of sloppy design directions, we conducted self-assembly optimizations in which only the patch opening angle, the stiffest parameter, was allowed to vary. Such optimizations have a significantly higher failure rate, taking longer to converge or converging to a region where no

desired structures assemble. The sloppy directions are thus also crucial for the design process, and this work introduces Hessians as an important parameterization and visualization tool for design optimization *via* gradient descent. Detailed analysis of Hessians and their spectra, global loss function landscape shapes, and the possible connection between the sloppy modes in the design process and the floppy modes in the self-assembly dynamics stand out as key questions for future work.

V. CONCLUSION

In this paper, we demonstrate how to utilize a gradient-based method as a learning tool for designing soft materials. We uncovered a nontrivial relationship between optimized design parameters and their dependency on the desired assembled structure, systematically compared inverse-design results from stabilization and self-assembly, taking into account kinetics, and provided new methods to interrogate the relationships between potential design parameters for building blocks. These insights are beneficial for any self-limiting assembly, especially in places where *de novo* protein design is used for closed structure assembly, as our method can identify the precision of control needed for a given set of parameters.

ACKNOWLEDGMENTS

The authors thank Carl Goodrich, Ella M. King, Andrei A. Klishin, and Andrea Liu for helpful discussions. This work was supported by the National Science Foundation (NSF) AI Institute in Dynamic Systems under Grant No. CBET-2112085, and NSF Grant DMR-2418928. The technical support and advanced computing resources from University of Hawaii Information Technology Services – Research Cyberinfrastructure, funded in part by the NSF CC* awards #2201428 and #2232862 are gratefully acknowledged. The computational workflow and data management for this publication was supported by the signac [51] and signac-flow [52] data management framework.

DATA AVAILABILITY

Code to reproduce the results of the manuscript has been stored at Zenodo [53].

-
- [1] Z. Zhang and S. C. Glotzer, Self-assembly of patchy particles, *Nano letters* **4**, 1407 (2004).
 - [2] A. Giacometti, F. Lado, J. Largo, G. Pastore, and F. Sciortino, Effects of patch size and number within a simple model of patchy colloids, *The Journal of chemical physics* **132** (2010).
 - [3] F. Sciortino and E. Zaccarelli, Reversible gels of patchy particles, *Current Opinion in Solid State and Materials Science* **15**, 246 (2011).
 - [4] L. Rovigatti, J. Russo, and F. Romano, How to simulate patchy particles, *The European Physical Journal E* **41**, 1 (2018).

- [5] R. A. M. Kalapurakal, B. C. Rocha, and H. Vashisth, Self-assembly in an experimentally realistic model of lobed patchy colloids, *ACS Applied Bio Materials* (2023).
- [6] I. G. Johnston, A. A. Louis, and J. P. Doye, Modelling the self-assembly of virus capsids, *Journal of Physics: Condensed Matter* **22**, 104101 (2010).
- [7] H. C. Klein and U. S. Schwarz, Studying protein assembly with reversible brownian dynamics of patchy particles, *The Journal of chemical physics* **140** (2014).
- [8] N. Gnan, F. Sciortino, and E. Zaccarelli, Patchy particle models to understand protein phase behavior, in *Protein Self-Assembly: Methods and Protocols* (Springer, 2019) pp. 187–208.
- [9] S. Li, G. Tresset, and R. Zandi, From disorder to icosahedral symmetry: How conformation-switching subunits enable rna virus assembly, *Science advances* **11**, eady7241 (2025).
- [10] A. W. Paine, M. F. Hagan, and V. N. Manoharan, Disassembly of virus-like particles and the stabilizing role of the nucleic acid cargo, *The Journal of Physical Chemistry B* **129**, 1516 (2025).
- [11] Q. Chen, S. C. Bae, and S. Granick, Directed self-assembly of a colloidal kagome lattice, *Nature* **469**, 381 (2011).
- [12] X. Mao, Q. Chen, and S. Granick, Entropy favours open colloidal lattices, *Nature materials* **12**, 217 (2013).
- [13] L. Koehler, P. Ronceray, and M. Lenz, How do particles with complex interactions self-assemble?, *Physical Review X* **14**, 041061 (2024).
- [14] C. Karner and E. Bianchi, Partially bonded crystals: a pathway to porosity and polymorphism, *ACS nano* **19**, 5146 (2025).
- [15] Zhang, A. S. Keys, T. Chen, and S. C. Glotzer, Self-assembly of patchy particles into diamond structures through molecular mimicry, *Langmuir* **21**, 11547 (2005).
- [16] A. Neophytou, D. Chakrabarti, and F. Sciortino, Facile self-assembly of colloidal diamond from tetrahedral patchy particles via ring selection, *Proceedings of the National Academy of Sciences* **118**, e2109776118 (2021).
- [17] W. K. den Otter, M. R. Renes, and W. J. Briels, Asymmetry as the key to clathrin cage assembly, *Biophysical journal* **99**, 1231 (2010).
- [18] L. Feng, R. Dreyfus, R. Sha, N. C. Seeman, and P. M. Chaikin, Dna patchy particles, *Adv. Mater* **25**, 2779 (2013).
- [19] S. Liu, C. Zhang, L. Li, X. Deng, C. Hu, F. Yang, Q. Liu, and W. Tan, Organization of an artificial multicellular system with a tunable dna patch on a membrane surface, *Nano Letters* **24**, 433 (2023).
- [20] Z. Gong, T. Hueckel, G.-R. Yi, and S. Sacanna, Patchy particles made by colloidal fusion, *Nature* **550**, 234 (2017).
- [21] X. Zheng, M. Liu, M. He, D. J. Pine, and M. Weck, Shape-shifting patchy particles, *Angewandte Chemie* **129**, 5599 (2017).
- [22] R. Niu, C. X. Du, E. Esposito, J. Ng, M. P. Brenner, P. L. McEuen, and I. Cohen, Magnetic handshake materials as a scale-invariant platform for programmed self-assembly, *Proceedings of the National Academy of Sciences* **116**, 24402 (2019).
- [23] Z. Liang, M. X. Lim, Q.-Z. Zhu, F. Mottes, J. Z. Kim, L. Gutteries, C. Smart, T. Pearson, C. X. Du, M. Brenner, *et al.*, Magnetic decoupling as a proofreading strategy for high-yield, time-efficient microscale self-assembly, *Proceedings of the National Academy of Sciences* **122**, e2502361122 (2025).
- [24] D. Hayakawa, T. E. Videbaek, D. M. Hall, H. Fang, C. Sigl, E. Feigl, H. Dietz, S. Fraden, M. F. Hagan, G. M. Grason, *et al.*, Geometrically programmed self-limited assembly of tubules using dna origami colloids, *Proceedings of the National Academy of Sciences* **119**, e2207902119 (2022).
- [25] D. Hayakawa, T. E. Videbæk, G. M. Grason, and W. B. Rogers, Symmetry-guided inverse design of self-assembling multiscale dna origami tilings, *ACS nano* **18**, 19169 (2024).
- [26] Z. Li, S. Wang, U. Nattermann, A. K. Bera, A. J. Borst, M. Y. Yaman, M. J. Bick, E. C. Yang, W. Sheffler, B. Lee, *et al.*, Accurate computational design of three-dimensional protein crystals, *Nature Materials* **22**, 1556 (2023).
- [27] A. Trubiano and M. F. Hagan, Markov state model approach to simulate self-assembly, *Physical Review X* **14**, 041063 (2024).
- [28] A. W. Long and A. L. Ferguson, Rational design of patchy colloids via landscape engineering, *Molecular Systems Design & Engineering* **3**, 49 (2018).
- [29] Y. Ma, J. C. Aulicino, and A. L. Ferguson, Inverse design of self-assembling diamond photonic lattices from anisotropic colloidal clusters, *The Journal of Physical Chemistry B* **125**, 2398 (2021).
- [30] S. Whitelam and I. Tamblyn, Neuroevolutionary learning of particles and protocols for self-assembly, *Physical review letters* **127**, 018003 (2021).
- [31] J. Russo, F. Romano, L. Kroc, F. Sciortino, L. Rovigatti, and P. Šulc, Sat-assembly: A new approach for designing self-assembling systems, *Journal of Physics: Condensed Matter* **34**, 354002 (2022).
- [32] C. Beneduce, F. Sciortino, P. Šulc, and J. Russo, Engineering azeotropy to optimize the self-assembly of colloidal mixtures, *ACS nano* (2023).
- [33] L. Y. Rivera-Rivera, T. C. Moore, and S. C. Glotzer, Inverse design of triblock janus spheres for self-assembly of complex structures in the crystallization slot via digital alchemy, *Soft Matter* **19**, 2726 (2023).
- [34] B. H.-Y. Wei, C. L. Petix, Q. Chen, M. P. Howard, and J. Mittal, Designing binary mixtures of colloidal particles with simple interactions that assemble complex crystals, *arXiv preprint arXiv:2507.13166* (2025).
- [35] S. S. Schoenholz and E. D. Cubuk, Jax, mdl: End-to-end differentiable, hardware accelerated, molecular dynamics in pure python, *arXiv preprint arXiv:1912.04232* (2019).
- [36] C. P. Goodrich, E. M. King, S. S. Schoenholz, E. D. Cubuk, and M. P. Brenner, Designing self-assembling kinetics with differentiable statistical physics models, *Proceedings of the National Academy of Sciences* **118**, e2024083118 (2021).
- [37] R. K. Krueger, E. M. King, and M. P. Brenner, Tuning colloidal reactions, *Physical Review Letters* **133**, 228201 (2024).
- [38] M. C. Engel, J. A. Smith, and M. P. Brenner, Optimal control of nonequilibrium systems through automatic differentiation, *Physical Review X* **13**, 041032 (2023).
- [39] E. M. King, C. X. Du, Q.-Z. Zhu, S. S. Schoenholz, and M. P. Brenner, Programming patchy particles for materials assembly design, *Proceedings of the National Academy of Sciences* **121**, e2311891121 (2024).

- [40] J.-P. Hansen, Phase transition of the lennard-jones system. II. high-temperature limit, **2**, 221, publisher: American Physical Society.
- [41] P. M. Morse, Diatomic molecules according to the wave mechanics. II. vibrational levels, **34**, 57, publisher: American Physical Society.
- [42] D. Kingma and J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (2014).
- [43] V. Ramasubramani, B. D. Dice, E. S. Harper, M. P. Spellings, J. A. Anderson, and S. C. Glotzer, freud: A software suite for high throughput analysis of particle simulation data, Computer Physics Communications **254**, 107275 (2020).
- [44] X. Mao and T. C. Lubensky, Maxwell lattices and topological mechanics, Annual Review of Condensed Matter Physics **9**, 413 (2018).
- [45] S. Auer and D. Frenkel, Suppression of crystal nucleation in polydisperse colloids due to increase of the surface free energy, Nature **413**, 711 (2001).
- [46] B. Luo, A. Kim, J. W. Smith, Z. Ou, Z. Wu, J. Kim, and Q. Chen, Hierarchical self-assembly of 3d lattices from polydisperse anisometric colloids, Nature communications **10**, 1 (2019).
- [47] C. P. Royall, P. Charbonneau, M. Dijkstra, J. Russo, F. Smallenburg, T. Speck, and C. Valeriani, Colloidal hard spheres: Triumphs, challenges, and mysteries, Reviews of Modern Physics **96**, 045003 (2024).
- [48] M. K. Transtrum, B. B. Machta, and J. P. Sethna, Geometry of nonlinear least squares with applications to sloppy models and optimization, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics **83**, 036701 (2011).
- [49] B. B. Machta, R. Chachra, M. K. Transtrum, and J. P. Sethna, Parameter space compression underlies emergent theories and predictive models, Science **342**, 604 (2013).
- [50] K. N. Quinn, M. C. Abbott, M. K. Transtrum, B. B. Machta, and J. P. Sethna, Information geometry for multiparameter models: New perspectives on the origin of simplicity, Reports on Progress in Physics **86**, 035901 (2022).
- [51] C. S. Adorf, P. M. Dodd, V. Ramasubramani, and S. C. Glotzer, Simple data and workflow management with the signac framework, Comput. Mater. Sci. **146**, 220 (2018).
- [52] V. Ramasubramani, C. S. Adorf, P. M. Dodd, B. D. Dice, and 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.
- [53] G. Snyder and C. X. Du, (2025).

Patchy Particles Design: From Floppy Modes to Sloppy Dimensions - Supplementary Information

Gregory Snyder¹ and Chrisy Xiyu Du^{1,*}

¹*Mechanical Engineering, University of Hawai'i at Mānoa, Honolulu, HI, 96822*

(Dated: October 13, 2025)

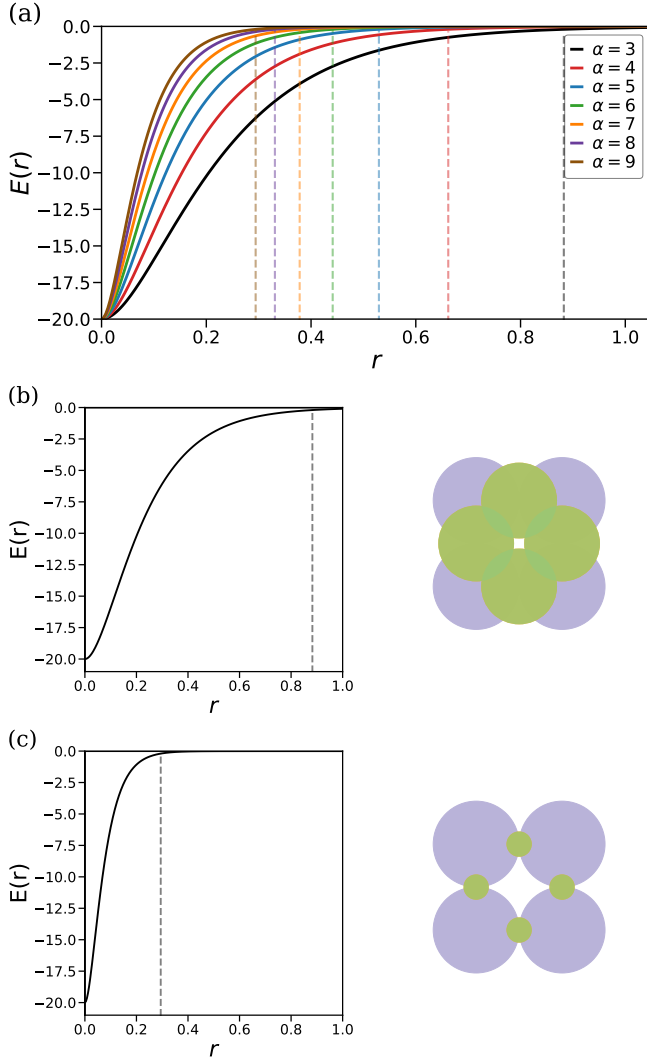


FIG. 1: (a) Morse potential for various α s with corresponding cut-off distances at $U_M(r_{\text{cut}}) = 0.01\epsilon$. As α increases, patch size decreases. (b-c) A side-by-side visualization of how changing the steepness of the Morse potential affects the patch size of the patchy particles. These figures correspond to $\alpha = [3, 9]$, respectively.

I. PATCH SIZE DETERMINATION

In this work, the patch size is determined by the steepness parameter of the Morse potential α . As α increases, patch size decreases. In Eq. 1,

$$U_M(r) = \epsilon_m(1 - e^{-\alpha(r-r_0)})^2, \quad (1)$$

we set $r_0 = 0$ so that binding is the strongest when two patches completely overlap. To provide an exact mapping between α and patch size, we set the cut-off distance r_{cut} to be $U_M(r_{\text{cut}}) = 0.01\epsilon$. By treating the bond distance as the diameter of the patch, we can rearrange Eq. 1 to solve for the patch radius below:

$$a = \frac{\ln(1 - \sqrt{0.99})}{-2\alpha} \quad (2)$$

II. BATCH SIZE BENCHMARK

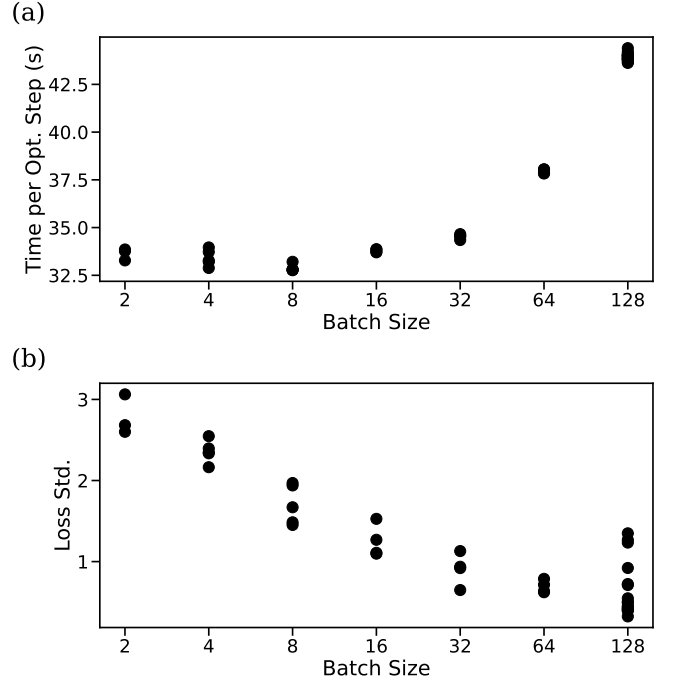


FIG. 2: (a) Computational time (in seconds) per optimization step with respect to different batch sizes for self-assembly optimizations. (b) Standard deviation of the loss at each optimization step with respect to different batch sizes for self-assembly optimizations.

* xiyudu@hawaii.edu

Since our self-assembly simulation is a stochastic process, we want to average over many simulations (batch sizes) with the same parameters when taking gradients. However, as we increase batch sizes, the total optimization time will increase, and we may also encounter out-of-memory errors during back propagation. To decide the optimal batch size for our optimizations, we tested a range of batch sizes from 2 to 128. We ran five independent optimizations at batch sizes of [2, 4, 8, 16, 32, 64, 128] on a NVIDIA H100 GPU [1, 2]. All benchmark simulations are done on the same type of GPU for consistency, and we chose H100 for its large memory (80 Gb) as JAX-MD is very GPU-memory intensive for large batch sizes.

The benchmark runs follow the same protocols described in the method section for self-assembly optimization. We chose a patch size corresponding to $\alpha = 5$, which was arbitrarily chosen. Fig. 2(a) shows the number of seconds for each optimization step at different batch sizes. We note that for batch sizes smaller than 20, there is no major time difference in an average optimization step. This results tell us that for our system, we can have a batch size of at least 16 for almost no additional time cost if the GPU memory permits. Fig. 2(b) shows the standard deviation of loss as a function of batch sizes. We note that the standard deviation plateaus after a batch size of 64, which is the size we ultimately chose for production runs.

For stabilization optimization, we decreased the batch size to 16, as the simulations are much less noisy, allowing for quicker access to other types of GPU nodes. Since the number of H100 on the University of Hawai'i's KOA HPC [3] is limited, we used NVIDIA A30[4], NVIDIA RTX A4000[5], NVIDIA Quadro RTX 5000[6], and Tesla V100-SXM2-32GB[7] when memory is not a huge concern.

III. STABILIZATION VERSUS SELF-ASSEMBLY

We provide additional comparisons between the self-assembly and stabilization optimization results for triangles and squares. For squares, we observe a notable overlap between the upper branch of the stabilization results and the self-assembly results. As the patch size decreases, the self-assembly results favor a slightly larger patch opening angle compared to stabilization; however, the number of data points is too low to be conclusive. For triangles, we plotted only the comparison between the morphology (ii) found in the stabilization optimization and the self-assembly optimization results. We note a nice overlap between the two.

These comparisons align well with discussions in other inverse-design strategies. Often, inverse-design methods can only search for optimal parameters that stabilize the final structure, while the kinetics of self-assembly cannot be accounted for [8, 9]. In our inverse-design framework, we can systematically investigate both scenarios

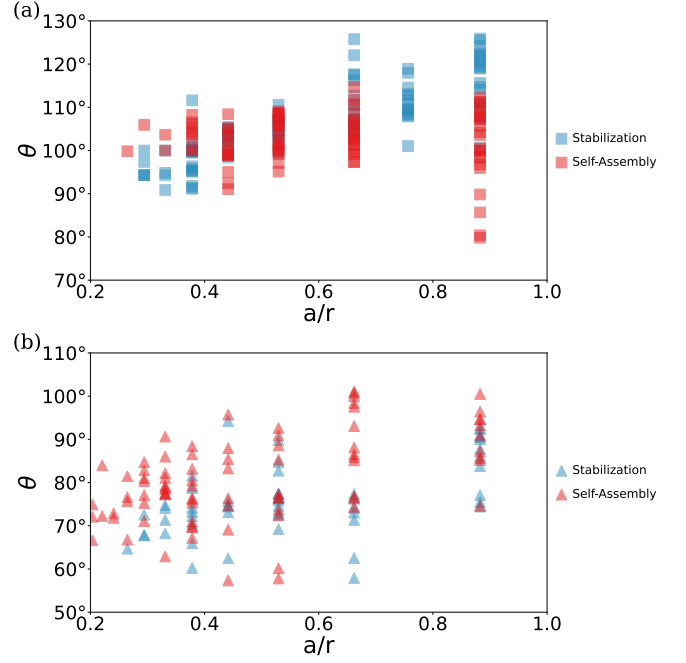


FIG. 3: Both figures display the optimal patch opening angles collected for various patch sizes in square and triangle optimizations, respectively. The red markers represent the results from self-assembly optimizations, while the blue markers represent the results from stabilization optimizations. (a) An overlapping comparison of square self-assembly and square stabilization high-opening-angle morphology. (b) An overlapping comparison of the triangle self-assembly and triangle stabilization desired morphology.

and identify the overlapping parameters. Here, we observe that self-assembly optimizations yield a subset of stabilization optimization parameters. Since stabilization optimizations are computationally much cheaper, we could limit the scope of the optimizer when examining more complex structures before performing self-assembly optimizations in the future. Additionally, in future work, we can explore the possibility of using parameters trained on stabilization optimizations as a starting point for self-assembly optimizations. This combined approach can significantly increase the efficiency of designing building blocks using our prescribed methodologies.

IV. BINDING ENERGY EVOLUTION

We provide a collection of 5 successful, self-assembly optimization's binding energy evolution for particles with a patch ratio of .53.

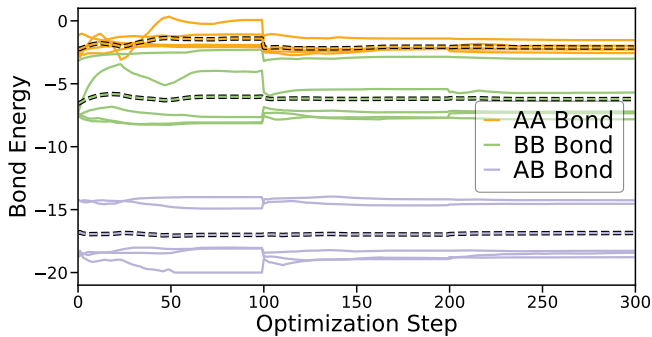


FIG. 4: A collection of 5 bond energy evolutions from successful self-assembly optimizations for particles with a patch ratio of 0.53. The dashed lines show the average bond energies at each optimization step.

V. SELF-ASSEMBLY TRAJECTORY SNAPSHOTS

We provide a small collection of square and triangle self-assembly's final snapshots using the optimal parameters found through the gradient-based optimization for different patch sizes. These simulations provide validation for our inverse-design methods.

VI. HESSIAN COMPUTATION FOR OPTIMIZATION PARAMETERS

We provide additional Hessian snapshots of the square and triangle self-assembly optimizations.

-
- [1] N. Corporation, Nvidia h100 tensor core gpu: Pcie specification, <https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/h100/pdf/nvidia-h100-datasheet.pdf> (2022), accessed: 2025-08-31.
 - [2] NVIDIA Corporation, Nvidia h100 nvl gpu: Product brief (pb-11773-001_v01), https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/h100/PB-11773-001_v01.pdf (2024), accessed: 2025-08-31.
 - [3] University of Hawai'i Information Technology Services, Research Cyberinfrastructure, Koa High Performance Computing (HPC) Cluster, <https://datascience.hawaii.edu/about-koa/> (2023), accessed: 2025-08-31.
 - [4] N. Corporation, NVIDIA A30 Tensor Core GPU Data Sheet, <https://www.nvidia.com/en-us/data-center/products/a30-gpu/> (2022), accessed: 2025-08-31.
 - [5] N. Corporation, NVIDIA RTX A4000 data sheet, <https://www.nvidia.com/content/dam/en-zz/Solutions/gtcs21/rtx-a4000/nvidia-rtx-a4000-datasheet.pdf> (2022), accessed: 2025-08-31.
 - [6] N. Corporation, NVIDIA Quadro RTX 5000 data sheet, <https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/quadro-product-literature/quadro-rtx-5000-us-nvidia-704076-r4-web.pdf> (2018), accessed: 2025-08-31.
 - [7] N. Corporation, NVIDIA Tesla V100 SXM2 32GB GPU data sheet, <https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-v100/nvidia-tesla-v100-datasheet.pdf> (2018), accessed: 2025-08-31.
 - [8] P. Zhou, J. C. Proctor, G. van Anders, and S. C. Glotzer, Alchemical molecular dynamics for inverse design, *Molecular Physics* **117**, 3968 (2019).
 - [9] B. H.-Y. Wei, C. L. Petix, Q. Chen, M. P. Howard, and J. Mittal, Designing binary mixtures of colloidal particles with simple interactions that assemble complex crystals, *arXiv preprint arXiv:2507.13166* (2025).
 - [10] V. Ramasubramani, B. D. Dice, E. S. Harper, M. P. Spellings, J. A. Anderson, and S. C. Glotzer, freud: A software suite for high throughput analysis of particle simulation data, *Computer Physics Communications* **254**, 107275 (2020).

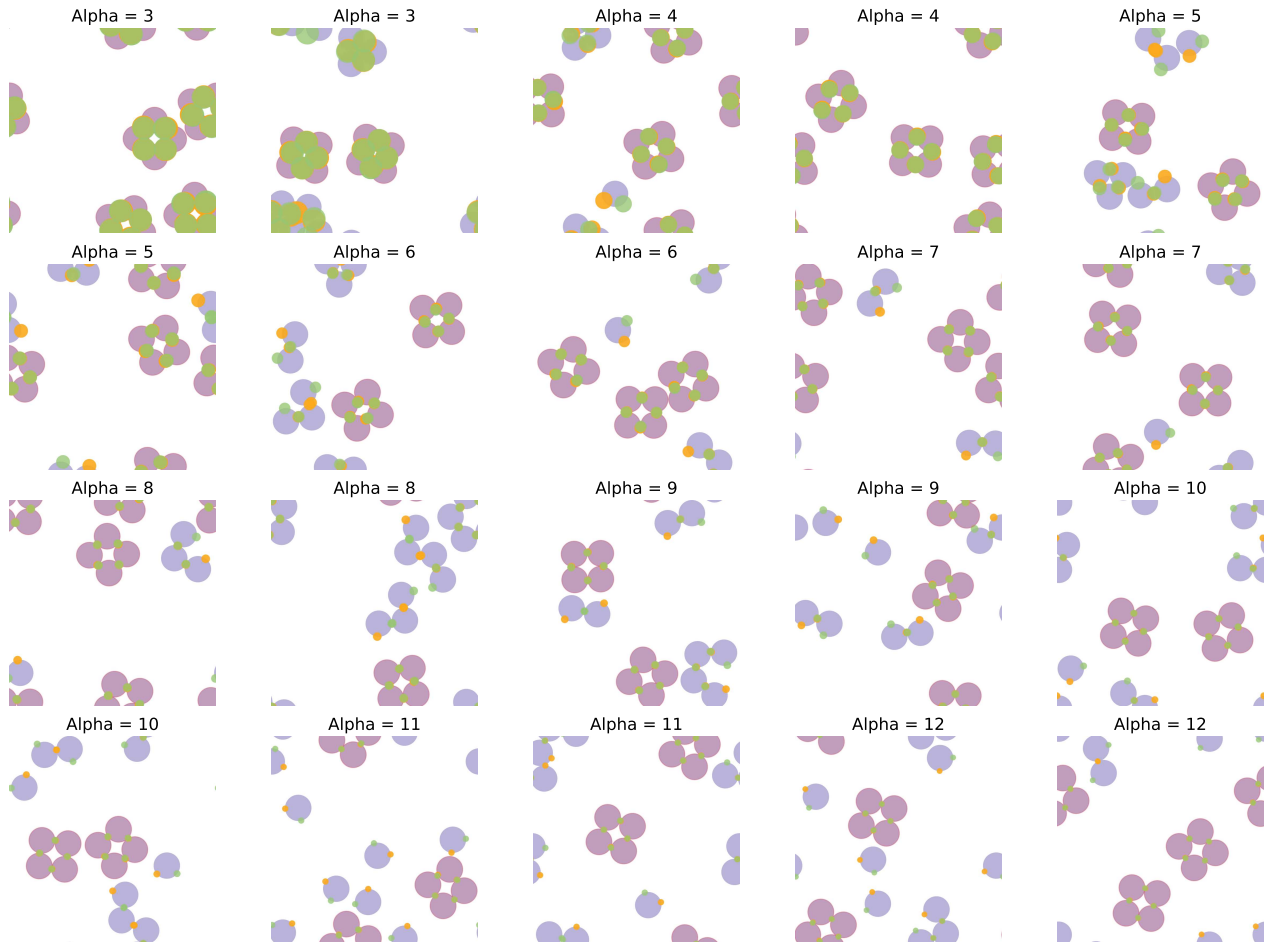


FIG. 5: A collection of 20 snapshots of the final state for square self-assembly with different patch sizes. Clusters that have been identified using freud [10] are highlighted.

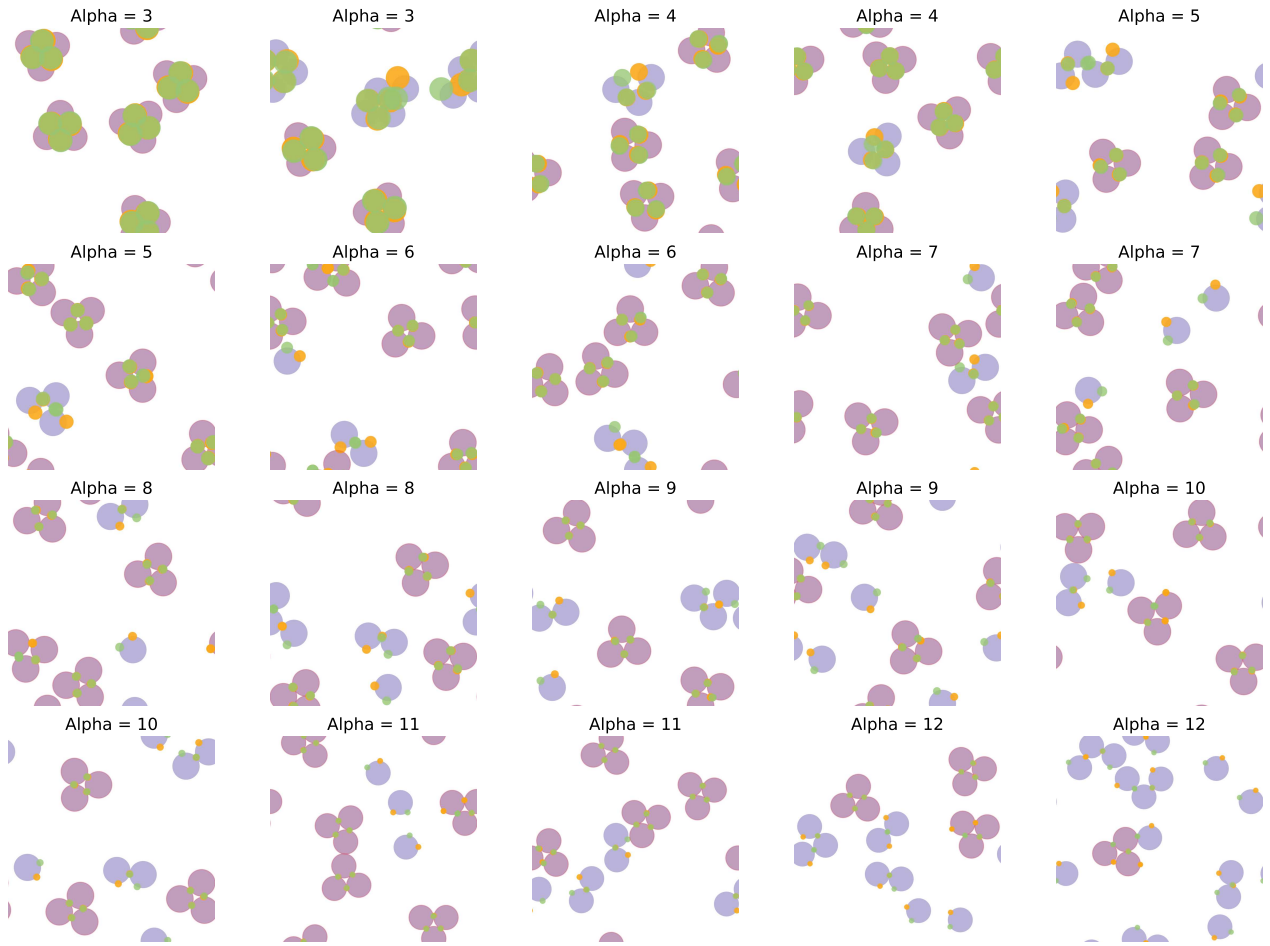


FIG. 6: A collection of 20 snapshots of the final state for triangle self-assembly with different patch sizes. Clusters that have been identified using freud are highlighted.

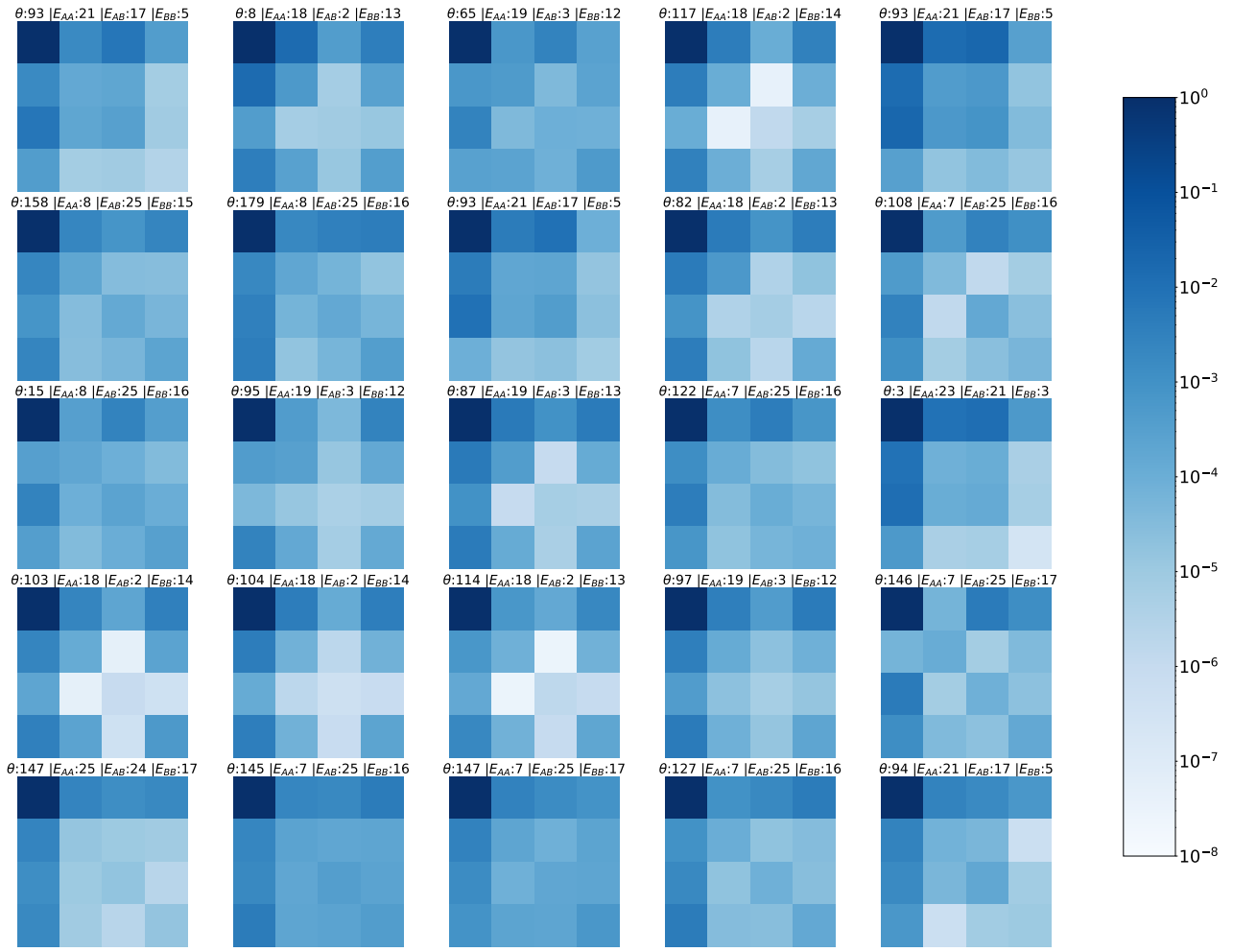


FIG. 7: A collection of 25 Hessians from snapshots of self-assembly optimizations of square structures.

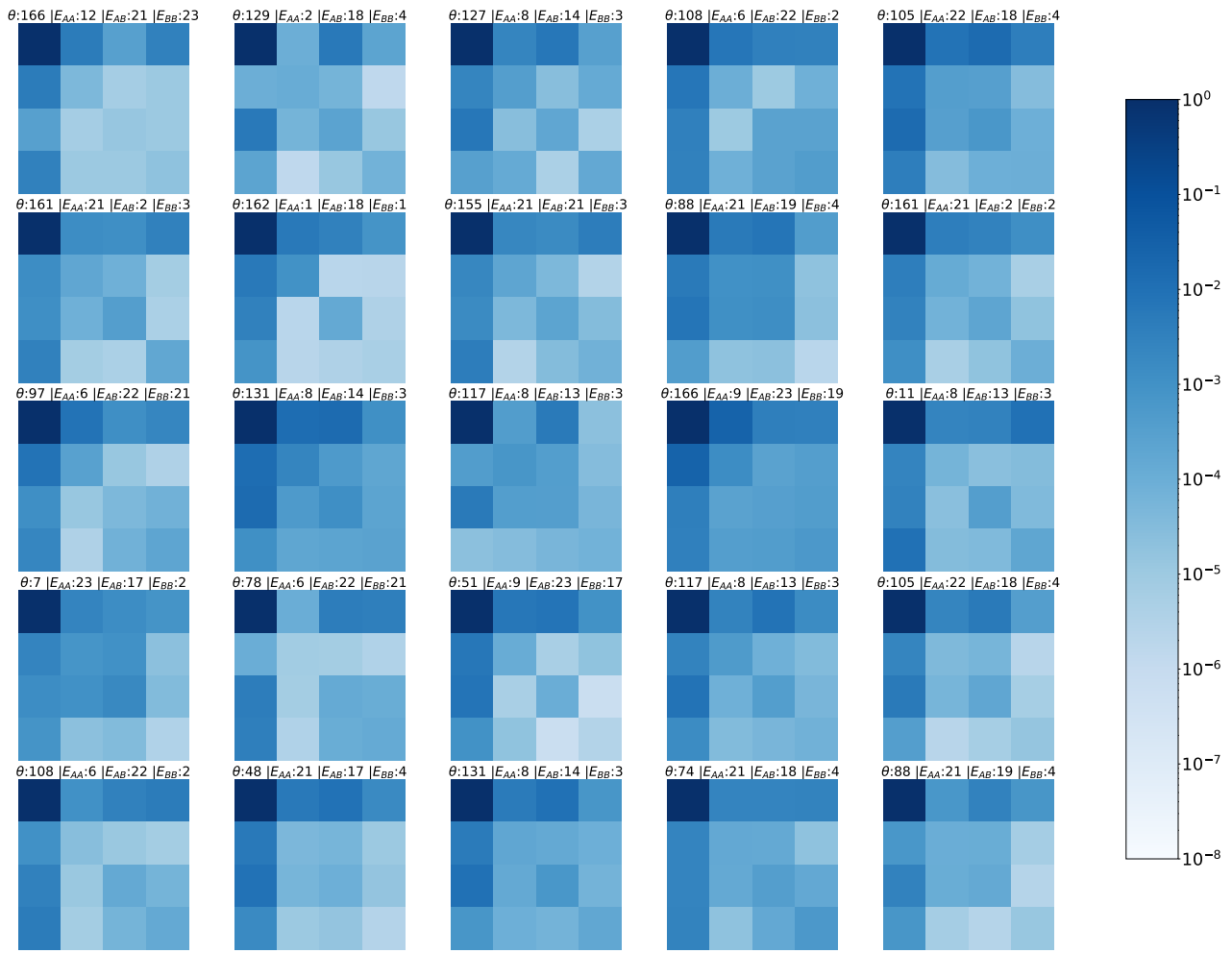


FIG. 8: A collection of 25 Hessians from snapshots of self-assembly optimizations of triangle structures.