Deep inverse design of hydrophobic patches on DNA origami for mesoscale assembly of superlattices

Po-An Lin[1] **Simiao Ren**[2] **Jonanthan Piland**[2]
Leslie M. Collins[2] **Stefan Zauscher**[1] **Yonggang Ke**[3] **Gaurav Arya**[1*]
[1]Dept. of Mechanical Engineering and Materials Science, Duke University
[2]Dept. of Electrical and Computer Engineering, Duke University
[3]Dept. of Biomedical Engineering, Georgia Institute of Technology and Emory University
[*]*Corresponding author*: gaurav.arya@duke.edu

Abstract

A major challenge in DNA nanotechnology is to extend the length scale of DNA structures from the nanoscale to the microscale to enable applications in cargo delivery, sensing, optical devices, and soft robotics. Self-assembly of DNA origami building blocks provides a promising approach for fabricating such higher-order structures. Inspired by self-assembly of patchy colloidal particles, researchers have recently begun to introduce patches of mutually attractive chemical moieties at designated sites on DNA origami to assemble them into complex higher-order architectures. However, designing such functionalized DNA origamis to target specific assembly structures is highly challenging because the underlying relationship between the building block design and assembly structure is very complex. Machine learning is especially well suited for such inverse-design tasks. In this work, we develop a coarse-grained model of DNA origami nanocubes grafted with hydrophobic brushes and employ the neural adjoint (NA) method to explore highly ordered target assemblies of such origamis, including checkerboard, honeycomb, and Kagome lattices. We envision that our design approach can be generalized to more complex designs and used to tailor structural properties to expand the application space of DNA nanotechnology.

1 Introduction

DNA nanotechnology, a field that has been growing rapidly for the past 40 years, harnesses the canonical base-pairing rules of DNA to program nanoscale structures through self-assembly (the spontaneous organization of matter). Of the techniques developed, DNA origami stands out a highly versatile approach to creating nanomaterials with unprecedented geometric precision complexity, high programmability, and functionality [1, 2, 3]. However, the size of these structures is generally confined to hundreds of nanometers, which greatly limits their throughput and functionality. Higher-order self-assembly of DNA origami offers a promising approach to achieving DNA materials in the micro- to macroscopic length scales. While giga-dalton DNA mesostructures have been achieved by site-specific base paring and by shape-complementary blunt-end stacking contacts, the design complexity and assembly of these structures is still formidable and typically results in low yields [4, 5].

Recently, we developed a surface-initiated enzyme-catalyzed polymerization approach to grow polynucleotide chains of natural and hydrophobic non-natural nucleotides at designated sites on DNA origami [6, 7]. This approach opens up a new, interesting avenue for self-assembling DNA origami structures, whereby patches of hydrophobic and hydrophilic brushes introduced at specific surface locations would endow the DNA origami with highly directional attractive and repulsive interactions.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

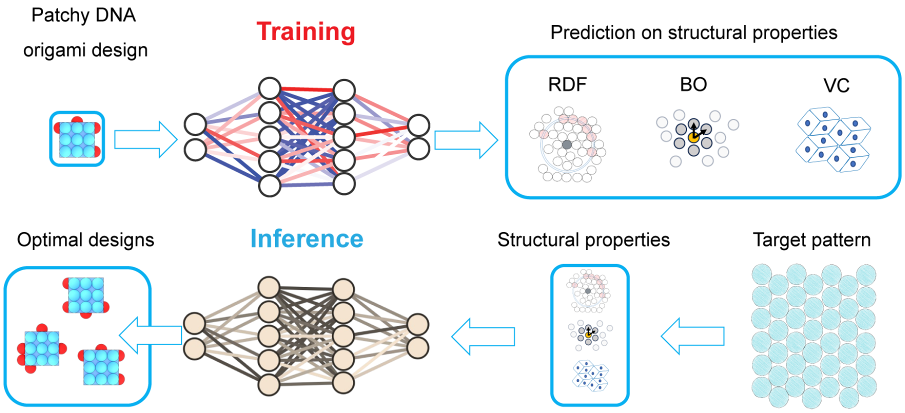

Figure 1: Overview of our DNA origami inverse-design strategy. The pipeline for creating optimal lattice designs involves extracting local structural properties such as radial distribution function (RDF), Voronoi cell area (VC), and bond orientational order parameters (BO) from simulations. The complied simulation dataset is used to train a neural adjoint method, which can identify optimal designs that produce similar properties for a given set of artificially created patterns.

We propose that our ability to introduce such precise "amphiphilic" interactions combined with the rich library of building block shapes attainable with DNA origami should enable the creation of truly unique DNA materials. However, finding the appropriate building block that would lead to a specific assembly architecture involves trial-and-error manipulation of the design variables of the assembly system, the length and location of the hydrophobic and hydrophilic patches and the solution temperature and ionic conditions in our case.

To facilitate this optimization process, "inverse-design" approaches are often adopted. Inverse design involves specifying a desired property and using an empirical or learned relationship between design and properties to deduce the appropriate design; The predicted design is then tested experimentally to confirm the prediction. Although several numerical optimization techniques have been employed, carefully crafted objective functions tailored to specific structural properties or applications are often required, and prior knowledge is rarely reused when objective functions are changed [8, 9, 10]. A viable alternative is deep learning. Deep neural networks (DNN) have successfully been used to learn design-property mappings in various design scenarios, including artificial electromagnetic materials [11], mechanical meta-surfaces [12] and drug discovery [13].

Even though deep learning has shown its promise in learning the forward process (i.e. from material design to property), the inverse problem has some unique challenges compared to forward deep learning methods, such as DNN with L_2 loss. The first is non-uniqueness, which arises from the backward mapping that is not present in the forward mapping. From physics, for each material design there exists a single property that can be attained, however, for each property there may exist multiple possible design matches, providing a confusing gradient to a vanilla neural network. The second is non-existence of a solution for some desired target property. The last one is discontinuity, where an infinitesimal change in the target structure space, leads to an enormous change in the design space. To address these issues, probabilistic models like conditional variational auto-encoder (cVAEs) [14], invertible neural networks (INNs) [15] and generative adversarial networks (GANs) [16] have been proposed, as well as optimization-based models like neural adjoint (NA) [17]. Previous benchmarks [18, 17] have demonstrated that while generative models are very effective at learning manifolds with high-dimensionality like natural images, their accuracy for material design is not as good as iterative methods such as NA.

In this work, we employed the NA method for the inverse design of DNA origami nanocubes (DONs) functionalized with hydrophobic patches. First, molecular dynamics (MD) simulations were carried out to collect training data and physical descriptors for our deep learning models. Then, we proposed several target periodic architectures and asked the deep inverse design algorithm to propose building block designs, namely length and location of the hydrophobic brushes on the DNA nanocube, that would self-assemble into the target structure (Fig. 1). We envision that this inverse design approach can be generalized to more complex designs and used to obtain novel structural properties that could expand the application space of DNA nanotechnology and colloidal self-assembly.

2 Related literature on inverse design

Optimization is a common approach for the inverse design of colloidal particles. Inverse design in such systems is particularly challenging because of the increased role of entropy in determining the globally stable assembly structures. Thus, commonly used objective functions like potential energy which is readily accessible are less useful, making it imperative to shift the focus to developing more suitable objective function[19, 20]. Several studies have successfully implemented inverse-design schemes using objective functions based on structural properties of the targeted assembly [21, 22, 19, 20]. For instance, Lindquist et al. used a maximum-likelihood-based method to search optimal interparticle potential parameters by comparing low-dimensional structural descriptors between the resulting assemblies during optimization and the target configuration [23]. However, challenges emerge when one can only access the target structure's high-dimensional description/representation, which cannot be readily translated into an objective function. A notable example would be quasicrystals, where the exact atom coordinates are often unknown, limiting the feasibility of specifying them using low-dimensional descriptors. A recent study formulated an objective function by first establishing the mapping between the target structure and the diffraction pattern (high dimensional image-based descriptor) using a convolutional neural network. Subsequently, covariance matrix adaptation evolution strategy (CMA-ES) was applied to identify new phases by adjusting parameters of interparticle interaction potentials and assembly conditions [24]. Another challenge in the optimization process is convergence to a solution that produces the target structure, but also other undesired competing structures. Instead of constructing an objective function based on the structural properties of the target configuration, a more efficient approach involves optimizing a 'proxy' objective function that ensures the formation of the target structure. This was recently demonstrated in a study using CMA-ES to discover design solutions that produce diamond lattices [9], where the yield of the target structure could be dramatically improved by maximizing the count of local motifs associated with the diamond lattice.

3 Methods

3.1 Simulation data collection

We carried out MD simulations of DONs functionalized with hydrophobic brushes to understand how DONs with different design parameters attract, assemble, and arrange into distinct structures on a 2D surface. In these simulations, we used a highly coarse-grained (CG) model of the DONs, where the solvent is treated implicitly and only the simplest geometric details of the DONs are preserved [25, 26]. This allowed us to explore the vast design space efficiently and access the large time and length scale associated with the assembly process, while still capturing the most salient structural features.

Specifically, the DONs were modeled as rigid squares made of a 3×3 grid of CG beads, where the size σ and mass m of each CG bead corresponds to a length scale of $\approx$10 nm and a molecular weight of $\approx$20,000 amu. Each patch of hydrophobic brushes, which collapses in a poor solvent condition to form a convex blob [27], was coarse-grained as an attractive bead of size = 1σ protruding from the surface of the rigid body. In this CG representation, there are two design variables: *patch location* and *protrusion ratio* (Fig. 2A). The patch location is a binary array with 12 elements (3 patches per edge and four edges per building block), where 1 and 0 indicate the presence or absence of a patch at each site. There are 1,043 unique patch locations for a 3×3 square after considering rotational symmetry (see Sec. 6.2 for more details). The protrusion ratio is a continuous variable representing the height of the hydrophobic brushes, expressed as d/D, where d is the distance the patch bead protrudes from the DON surface, and D is the diameter of the beads making up the DON (1σ). We considered the protrusion ratio to vary from 0.1 to 1.0 in increments of 0.1, and all patches on the DON were assumed to share the same protrusion ratio to reduce the design space.

For the force field, all intermolecular interactions other than hydrophobic attraction were modeled using an excluded-volume potential (Weeks-Chandler-Anderson or WCA potential) [28], and the hydrophobic attraction was treated using the Lennard-Jones potential (see Sec. 6.3 for details). For convenience, we considered the same size and energy parameters σ and ε for these potentials, where ε defines the energy scale of the system.

All simulations were carried out using LAMMPS [29] in the canonical (NVT) ensemble starting at a fixed temperature $T = 1.0\varepsilon/k_{\mathrm{B}}$, where k_{B} is the Boltzmann constant. A velocity-Verlet algorithm with a timestep of 0.01 τ and a Langevin thermostat with a damping constant of 100 τ^{-1} were used for integrating the equations of motion and controlling the temperature, where $\tau = \left(m\sigma^2/\varepsilon\right)^{1/2}$ is the characteristic time unit. We set up a simulation system with enough DONs to capture the assembly process within reasonable computational times. Each simulation involved 400 CG DONs at a low number density of 0.04 in a 2D simulation box with periodic boundary conditions, initiated from randomly dispersed DONs. To access low-energy or ground-state assemblies, we cooled the simulation according to the following annealing protocol: from $T = 1.0$ to $T = 0.3$ for 10^6 timesteps and from $T = 0.3$ to $T = 0.0$ for 5×10^7 timesteps with a large timestep size of 0.01τ. Then, we computed the structural descriptors of the assemblies based on the coordinates of the centers of each building block from the final configuration obtained at the end of the simulation run.

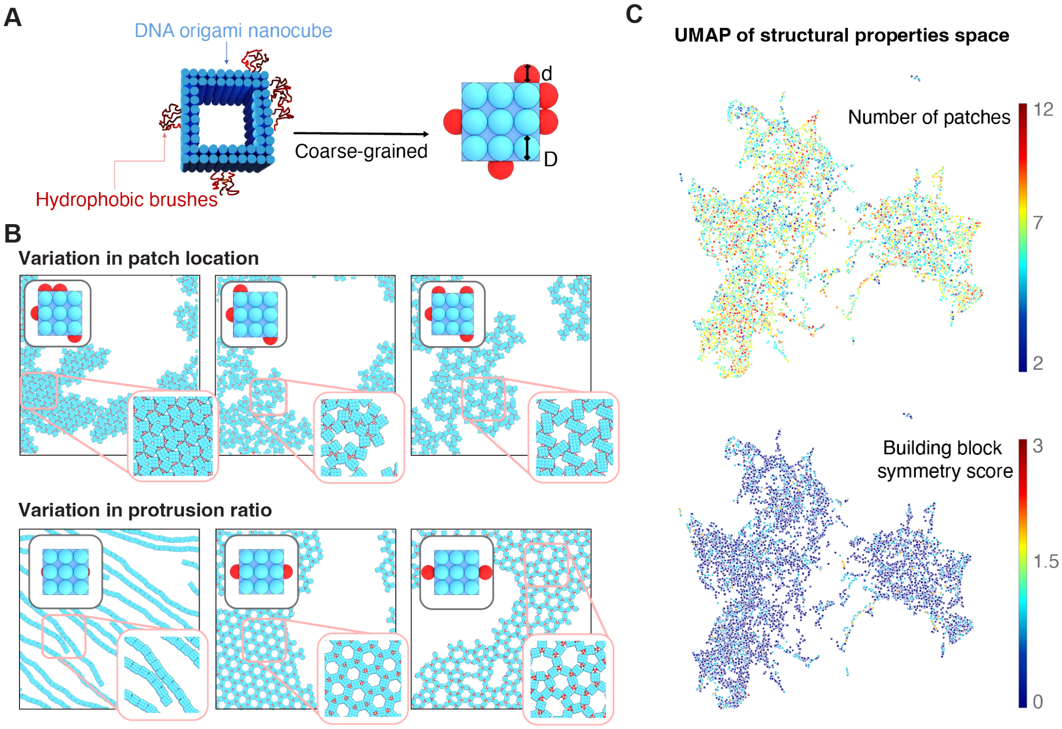

Figure 2: Coarse-grained model of a patchy DNA origami nanocube and its complex design-structure relationship. (A) The DNA origami nanocube with hydrophobic patches are coarse-grained as a 2D square with two design variables: patch location and protrusion ratio d/D. (B) Small variations in both design variables could lead to significant changes in the resulting assemblies, with their close-up view and corresponding building blocks shown in the bottom right and top left insets respectively. Top panel: starting with a given design (left), removal (middle) or displacement (right) of the patch results in three entirely different assemblies. Bottom panel: an increase in the protrusion ratio of a given design transforms the assemblies from 1D string to a honeycomb lattice, then to a defective structure with pentamer and hexamer motifs. (C) Basic design indicators, such as the number of patches (top) and symmetry of building blocks (bottom), highlight no distinct phase regions within the UMAP representation of structural properties and exhibit minimal correlation with them.

3.2 Simulation data annotation

From the 10,430 DON designs explored, a miscellaneous array of assembled structures emerged: ordered or disordered, elementary or hierarchical, finite-sized or lattice, close-packed or open. To characterize these structures, we adopted three informative descriptors that indicate their local ordering: radial distribution function (RDF), Voronoi cell area (VC), and bond orientation order (BO). We used an in-house code to compute RDFs, and the Freud package [30] to compute VCs and BOs. The RDF describes the local environment (particle arrangement) by quantifying how neighboring particles distribute around a particle. On the other hand, VC represents the space enclosed by neighboring particles. For instance, close-packed and open lattices have distinct VCs. BO reflects the local angular symmetry, and we select symmetries from two- to eight-fold to capture the local angular distribution of particles.

However, due to the presence of polymorphs (energetically similar but structurally different patterns), defects, and frustration in simulations, the descriptors are noisy and, therefore, require further processing. To address this, we computed distributions in VC and BO, instead of their ensemble-averaged value. This approach is preferred as averaging over multimodal distribution results in loss of valuable information. Additionally, we only selected portions of the RDF as our input data. Due to the stochasticity of our simulations, it is hard to observe the apparent long-range order. Consequently, a large portion of RDF is not informative, and it brought in significant errors when we inferred the desired design using artificially created crystal lattice patterns, which have strong and apparent long-range order. To account for this limitation, we set a cutoff of 18.75σ for the RDF descriptor. The variance of physical descriptors is shown in Sec. 6.7.

3.3 Inverse design method

We used NA to tackle our inverse design problem, which consists of two stages: (i) training: the NA method first trains a neural network to approximate the design-structure mapping f. (ii) inference: the design solution $\hat{x}$ is inferred by fixing weights and biases of $\hat{f}$, and computing the forward model's gradient with respect to the input of the network (i.e., the design). Specifically, the training stage involves training a neural network to predict the structural descriptors based on the input design, resulting in $\hat{f}$, an approximation of the forward model. In the inference stage, given the descriptors of a target structure y, NA iteratively adjusts its estimated solution (starting from randomly chosen designs x_0) using $\partial\hat{f}/\partial x$. Notably, calculation of $\partial\hat{f}/\partial x$ is straightforward as $\hat{f}$ is in a closed-form differentiable expression. Moreover, modern deep learning packages can be used to estimate gradients efficiently given a loss function $\mathcal{L}$ (mean squared error (MSE) in our cases). Thus, if y and x_i are the target output and current estimate of the solution respectively, one can compute x_{i+1} *via*

$$\hat{x}_{i+1} = \hat{x}_i - \alpha \left. \frac{\partial \mathcal{L}\left(\hat{f}\left(\hat{x}_i\right), y\right)}{\partial x} \right|_{x=\hat{x}_i} \tag{1}$$

where α is the learning rate. $\hat{x}_{i+1}$ can then be evaluated iteratively until some convergence criteria is satisfied (e.g., $\mathcal{L}$ changes very little after each iteration). Our initial solution, $\hat{x}_0$ is drawn from some distribution Γ. Given some desired y, NA iteratively adjusts its estimated solution (beginning with $\hat{x}_0$) until convergence (e.g., $\mathcal{L}$ no longer reduces). This entire process acts as the inverse model for the process, $f^{-1}(y, z)$, where $z = x_0$.

3.4 Training procedure

We split 95% of our simulation data into a training set and a validation set using an 80/20% ratio and used the rest of the 5% as a held-out test set. The input variable is naturally limited within the range of [0, 1], and the output physical descriptors were normalized per dimension to have a range of [0, 1]. During the training, we incorporated data augmentation techniques to facilitate the model training by exploiting the rotational symmetry of our system, as shown in Sec.6.6. The detailed architecture of the neural network is shown in Sec.6.5.

3.5 Evaluation procedure

To test our inverse design algorithm, we input physical descriptors of target ideal lattices and asked the NA method to output designs corresponding to these ideal lattice types. The target lattice we chose was (i) Kagome lattice (ii) hexagonal lattice and (iii) square/checkerboard lattice. For each target, we asked the NA to output the top 10 design configurations. In a usual inverse design paradigm, re-simulations would usually be carried out for the selected designs after this step. However, since we have a dense enough dictionary of simulated structures, we instead look for the nearest neighbor of the proposed designs in our simulation dictionary and compare their results.

Regarding the evaluation process of our method, since NA was originally designed to optimize continuous-space geometry parameters, and our design space is largely discrete (due to the binary indicator functions for hydrophobic patch locations), we need to modify the original NA implementation for our application. Previous attempts of modifying NA to discrete cases include training a VAE to encode the discrete space into low dimensional continuous latent space [31]. As our discrete space is relatively small with 2^{14} possible combinations, by using modern GPU resources with large RAM, we were able to initialize multiple geometric configurations for all possible configurations during a single inference step. Through this implementation, we circumvented the extra error that could be introduced during the training of an extra VAE encoder.

4 Results and discussion

4.1 Simulation data

We conducted a total of 10,430 simulations to sample the entire design space of patchy DON building blocks. The simulation results indicate that the underlying design-structure relationship/landscape is complex and non-smooth, where slight perturbations in the design space drive the assembly to very different structures. For instance, adding or displacing a patch on a building block or increasing the protrusion ratio leads to completely different assemblies, as depicted in Fig. 2. Next, we tested whether simple design rules could describe the design-assembly relationship. To this end, we implemented several dimensionality reduction approaches on the structural descriptors of all obtained assemblies (see Sec. 6.4). We then annotated the dimension-reduced structural space based on design indicators, such as the number of patches and the symmetry score of the building blocks, as defined in Sec. 6.1. We found out that these basic indicators cannot be used to identify assembly phases and are poorly correlated with the structural descriptors. This implies that it is difficult to infer the assembly morphology from the design simply based on intuition, and a deep learning model may be needed for inverse design.

4.2 Neural-adjoint training results

Using procedures described earlier, we implemented the NA method on our simulation dataset, exploring several different network architectures. Our model complexity hyper-parameter sweeping results are summarized in Fig. 3A and 3B. When we used all three descriptor sets in our training process, the problem complexity is best served with six layers of 1,000 neurons. In contrast, upon removing BO from our descriptor set, the model achieved optimal performance with nine layers of 1,000 neurons. We expected that increasing the number of descriptors would elevate the problem difficulty as more predictions are required. However, the results presented in Fig. 3A and 3B suggest otherwise. We propose that the additional information provided by the BO descriptor may enable the network to learn with fewer trainable parameters.

We further conducted an ablation study on the descriptors (i.e. removing descriptors systematically and estimating their best model validation error), and the results are shown in Table 1. We find that the combination of two descriptors might be more informative to the model as the MSE of double descriptors is lower than the averaged single-descriptor MSE. The combination of all descriptors, however, yields slightly worse performance. This adversarial effect may arise because, at this stage, an additional descriptor introduces more problem complexity with less information gain. Apart from the single-value MSE, we also provide both the distribution of the error made by our forward model trained in Fig. 3C and some example fits made by our model in Fig. 3D. From Fig. 3D, we observe that a low MSE indeed corresponds to a good model fit that effectively captures most of the variations in the descriptors. This lays a solid foundation for the subsequent inverse design step.

4.3 Inverse design results

Lastly, we used our top-performing NA model architectures to obtain building-block designs for three different candidate assembly structures, input into the algorithm based on their structural descriptors. We found that the model trained on two descriptors (VC and BO) provided the best inverse-design

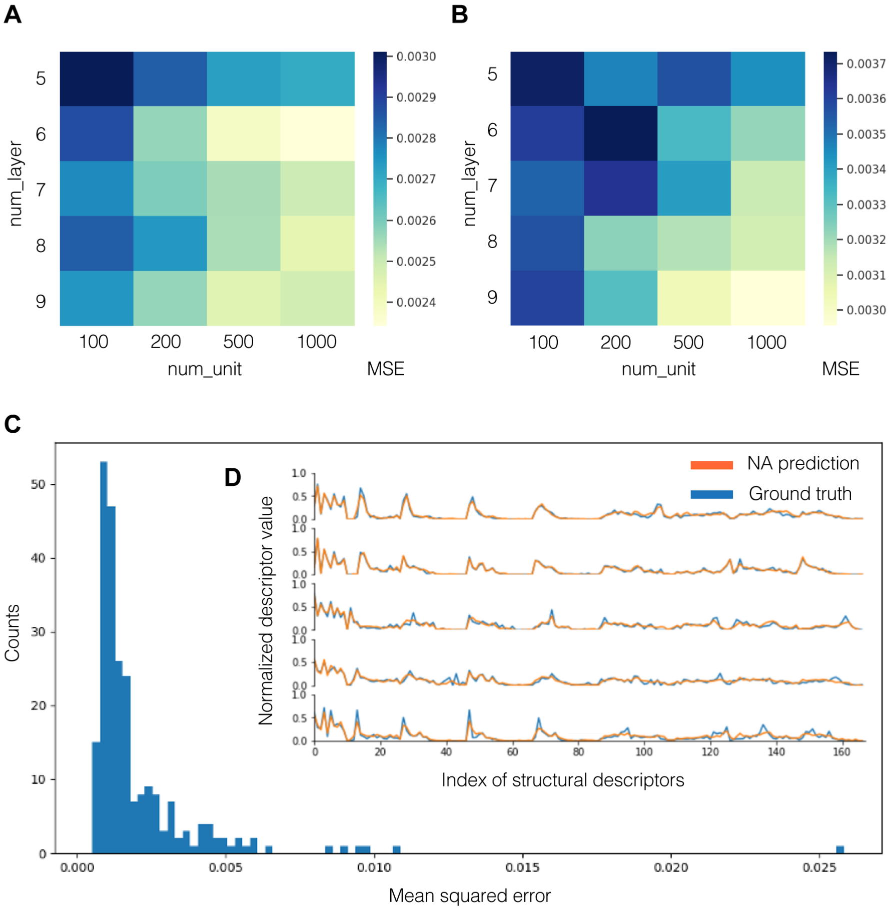

Figure 3: Training results of NA. (A,B) Model complexity hyper-parameter sweeping results. Color map represents validation loss when (A) all three descriptor sets were used (RDF, VC, and BO) and (B) only RDF and VC were used. (C) Histogram of the error made on the held-out test set for the model with all descriptors (labels). The average MSE on this set is 2.02×10^{-3}, which is comparable to our reported validation loss in Table 1. (D) Example predictions on the held-out test set for the model with all the physical descriptors. Both the ground truth and the predicted results are shown. The first 10 indices are RDF, while indices 10–27 are VC, and the remaining indices are BO.

Table 1: Ablation study on descriptors.

Descriptor sets		MSE value
Single descriptor	RDF	3.05×10^{-3}
	VC	2.96×10^{-3}
	BO	2.15×10^{-3}
Double descriptors	RDF+VC	2.95×10^{-3}
	VC+BO	2.19×10^{-3}
	RDF+BO	2.17×10^{-3}
All descriptors	RDF+VC+BO	2.34×10^{-3}

results, as shown in Fig 4. Overall, the assemblies of the designs predicted by NA match well with the target structures. Specifically, when we employed the NA to create a square/checkerboard lattice, the outcome yielded well-ordered structures composed of cubic motifs (top panel in Fig. 4). Furthermore, it is worth noting that the model's predictions are not limited to locally close-packed structures. It also predicted fascinating open checkerboard lattices with hierarchical tetramer motifs (middle and right pictures in the top panel of Fig. 4). Identifying the building blocks of lattice structures with hierarchical motifs presents a challenge due to their multi-scale nature. In the case of hexagonal lattices (middle panel in Fig. 4), we obtained hexagonally close-packed structures, which reasonably match our expectations. The model prediction becomes dramatic, however, when it comes to inverse-designing the Kagome lattices (bottom panel in Fig. 4). While the model did manage to find the correct design for this lattice, it also proposed off-target designs characterized by high porosity (large VC) in their simulated assemblies. It is apparent that the model's performance for the Kagome lattice becomes worse compared to targeting the simpler checkerboard or hexagon patterns. We hypothesize that it is because the Kagome lattice is a rare structure in the entire assembly phase space. Therefore, the models were not trained with enough examples having characteristics of Kagome patterns, and the uncertainty of models is reflected in the predicted designs. Another reason for obtaining sub-optimal design solutions is our use of a fixed nearest-neighbor interparticle distance of 3σ to generate candidate lattices. Ideally, lattices with different interparticle distances should be used as design solutions may not exist for a large fraction of the inputted lattice dimensions. In summary, the designs suggested by the NA method were found to have corresponding equivalents in the CG system, such as the checkerboard lattice, hexagonal lattice, and Kagome lattice. This demonstrates the flexibility and generalizability of the NA method, where only the pattern is required to be specified.

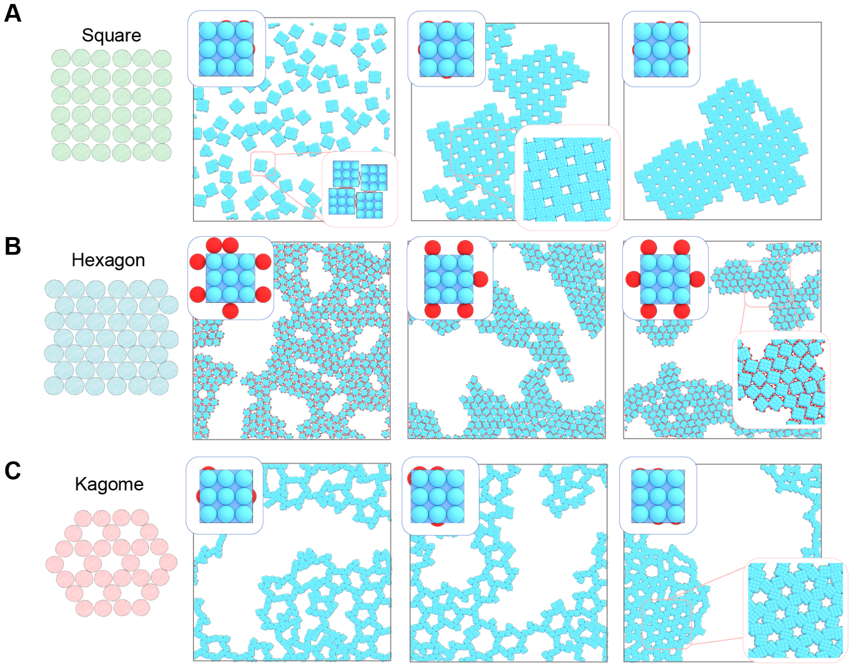

Figure 4: Target crystal lattices and inverse-design solutions. The patterns are (A) square lattice, (B) hexagonal lattice, and (C) Kagome lattice with their simulated inverse-design assemblies shown in the same panel. The proposed designs and the close-up view of their resulting assemblies are also provided in the top left and the bottom right insets.

5 Conclusions

In this work, we employed the NA method to predict the arrangement of hydrophobic brushes on DNA origami building blocks based on input structural properties of their desired assemblies and demonstrated that the algorithm can accurately determine designs that produce the desired structures.

Finding the correct building block design is extremely challenging, primarily due to the complex and non-smooth nature of the underlying design-structure relationship landscape. Basic design indicators, such as building block symmetries or number of patches, cannot be used to pinpoint desired phase regions within the structural properties space, as demonstrated by our dimensionality reduction analysis. Nevertheless, the NA method adopted here can discover the arrangement of hydrophobic brushes on DNA origami nanocubes required to assemble into Kagome lattices, hexagonal lattices, and checkerboard lattices. Moreover, the method also found the hierarchical equivalents of the target structures, which is difficult to achieve using the optimization approach. We envision that the NA method could be useful for the inverse design of more complex systems, such as amphiphilic DNA origami and DNA origami with multiple brush lengths on a single structure. Our future plans include exploring additional shapes of DNA origami as well as aperiodic patterning scenarios using the NA method to target the assembly of more complex functional materials, enabling a new range of applications of DNA nanotechnology.

References

[1] Paul W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, March 2006.

[2] Wei Sun, Etienne Boulais, Yera Hakobyan, Wei Li Wang, Amy Guan, Mark Bathe, and Peng Yin. Casting inorganic structures with DNA molds. *Science*, 346(6210):1258361, November 2014.

[3] Jakob Bach Knudsen, Lei Liu, Anne Louise Bank Kodal, Mikael Madsen, Qiang Li, Jie Song, Johannes B. Woehrstein, Shelley F. J. Wickham, Maximilian T. Strauss, Florian Schueder, Jesper Vinther, Abhichart Krissanaprasit, Daniel Gudnason, Anton Allen Abbotsford Smith, Ryosuke Ogaki, Alexander N. Zelikin, Flemming Besenbacher, Victoria Birkedal, Peng Yin, William M. Shih, Ralf Jungmann, Mingdong Dong, and Kurt V. Gothelf. Routing of individual polymers in designed patterns. *Nature Nanotechnology*, 10(10):892–898, October 2015.

[4] Grigory Tikhomirov, Philip Petersen, and Lulu Qian. Fractal assembly of micrometre-scale DNA origami arrays with arbitrary patterns. *Nature*, 552(7683):67–71, December 2017.

[5] Klaus F. Wagenbauer, Christian Sigl, and Hendrik Dietz. Gigadalton-scale shape-programmable DNA assemblies. *Nature*, 552(7683):78–83, December 2017.

[6] Yunqi Yang, Qinyi Lu, Chao-Min Huang, Hongji Qian, Yunlong Zhang, Sonal Deshpande, Gaurav Arya, Yonggang Ke, and Stefan Zauscher. Programmable Site-Specific Functionalization of DNA Origami with Polynucleotide Brushes. *Angewandte Chemie*, 133(43):23429–23435, October 2021.

[7] Yunqi Yang, Qinyi Lu, Yu Chen, Marcello DeLuca, Gaurav Arya, Yonggang Ke, and Stefan Zauscher. Spatiotemporal Control over Polynucleotide Brush Growth on DNA Origami Nanostructures. *Angewandte Chemie International Edition*, 62(48):e202311727, 2023. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.202311727.

[8] D. Chen, G. Zhang, and S. Torquato. Inverse Design of Colloidal Crystals via Optimized Patchy Interactions. *The Journal of Physical Chemistry B*, 122(35):8462–8468, September 2018. Publisher: American Chemical Society.

[9] Andrew L. Ferguson, Yutao Ma, and Joseph C. Aulicino. Inverse design of self-assembling diamond photonic lattices from anisotropic colloidal clusters. *Journal of Physical Chemistry B*, 125(9):2398–2410, March 2021. Publisher: American Chemical Society.

[10] Andrew W Long. Rational design of patchy colloids via landscape engineering. page 18, 2018.

[11] Y Deng, S Ren, K Fan, JM Malof, and WJ Padilla. Neural-adjoint method for the inverse design of all-dielectric metasurfaces. *Optics Express*, 29(5):7526–7534, 2021.

[12] HT Kollmann, DW Abueidda, S Koric, E Guleryuz, and NA Sobh. Deep learning for topology optimization of 2D metamaterials. *Materials and Design*, 196, 2020.

[13] R Gupta, D Srivastava, M Sahu, S Tiwari, RK Ambasta, and P Kumar. Artificial intelligence to deep learning: Machine intelligence approach for drug discovery. *Molecular Diversity*, 25(3):1315–1360, 2021.

[14] DP Kingma and M Welling. Auto-encoding variational bayes. 2014. arXiv: arXiv:1312.6114 [cs.LG].

[15] L Ardizzone, J Kruse, C Rother, and U Köthe. Analyzing inverse problems with invertible neural networks. In *International conference on learning representations*, 2019.

[16] I Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, A Courville, and Y Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, volume 27, pages 2672–2680, 2014.

[17] S Ren, W Padilla, and J Malof. Benchmarking deep inverse models over time, and the neural-adjoint method. In *Advances in neural information processing systems*, volume 33, pages 38–48, 2020.

[18] S Ren, A Mahendra, O Khatib, Y Deng, WJ Padilla, and JM Malof. Inverse deep learning methods and benchmarks for artificial electromagnetic material design. *Nanoscale*, 14(10):3958–3969, 2022.

[19] Marjolein Dijkstra and Erik Luijten. From predictive modelling to machine learning and reverse engineering of colloidal self-assembly. *Nature Materials*, 20(6):762–773, June 2021.

[20] Zachary M. Sherman, Michael P. Howard, Beth A. Lindquist, Ryan B. Jadrich, and Thomas M. Truskett. Inverse methods for design of soft materials. *The Journal of Chemical Physics*, 152(14):140902, April 2020.

[21] R. B. Jadrich, J. A. Bollinger, B. A. Lindquist, and T. M. Truskett. Equilibrium cluster fluids: pair interactions via inverse design. *Soft Matter*, 11(48):9342–9354, December 2015. Publisher: The Royal Society of Chemistry.

[22] Mikael C. Rechtsman, Frank H. Stillinger, and Salvatore Torquato. Optimized Interactions for Targeted Self-Assembly: Application to a Honeycomb Lattice. *Physical Review Letters*, 95(22):228301, November 2005. Publisher: American Physical Society.

[23] Beth A. Lindquist, Ryan B. Jadrich, and Thomas M. Truskett. Inverse Design for Self Assembly via On-the-Fly Optimization. *The Journal of Chemical Physics*, 145(11):111101, September 2016. arXiv:1609.00851 [cond-mat].

[24] Gabriele M. Coli, Emanuele Boattini, Laura Filion, and Marjolein Dijkstra. Inverse design of soft materials via a deep learning–based evolutionary strategy. *Science Advances*, 8(3):eabj6731, January 2022. Publisher: American Association for the Advancement of Science.

[25] Thomas E. Gartner and Arthi Jayaraman. Modeling and Simulations of Polymers: A Roadmap. *Macromolecules*, 52(3):755–786, February 2019.

[26] Marcello DeLuca, Sebastian Sensale, Po-An Lin, and Gaurav Arya. Prediction and Control in DNA Nanotechnology. *ACS Applied Bio Materials*, page acsabm.2c01045, March 2023.

[27] Yufeng Wang, Andrew D. Hollingsworth, Si Kyung Yang, Sonal Patel, David J. Pine, and Marcus Weck. Patchy Particle Self-Assembly via Metal Coordination. *Journal of the American Chemical Society*, 135(38):14064–14067, September 2013. Publisher: American Chemical Society.

[28] John D. Weeks, David Chandler, and Hans C. Andersen. Role of Repulsive Forces in Determining the Equilibrium Structure of Simple Liquids. *The Journal of Chemical Physics*, 54(12):5237–5247, June 1971.

[29] Aidan P. Thompson, H. Metin Aktulga, Richard Berger, Dan S. Bolintineanu, W. Michael Brown, Paul S. Crozier, Pieter J. in 't Veld, Axel Kohlmeyer, Stan G. Moore, Trung Dac Nguyen, Ray Shan, Mark J. Stevens, Julien Tranchida, Christian Trott, and Steven J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271:108171, February 2022.

[30] Vyas Ramasubramani, Bradley D. Dice, Eric S. Harper, Matthew P. Spellings, Joshua A. Anderson, and Sharon C. Glotzer. freud: A software suite for high throughput analysis of particle simulation data. *Computer Physics Communications*, 254:107275, September 2020.

[31] H Yu, S Ren, LM Collins, and JM Malof. Meta-simulation for the automated design of synthetic overhead imagery. 2022. arXiv: arXiv:2209.08685 [cs.CV].

[32] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[33] Y Wang, H Huang, C Rudin, and Y Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization. *arXiv preprint arXiv:2012.04456*, 2020.

[34] E Amid and MK Warmuth. TriMap: Large-scale dimensionality reduction using triplets. 2022. arXiv: arXiv:1910.00204 [cs.LG].

6 Supplementary Material

6.1 Building block symmetry score

The symmetry score presented in Fig. 2 C is based on the rotational symmetry of DON. In the case of a design with an identical sequence of patches on opposite sides (semi-symmetric design), it receives one point. Designs with twofold symmetry are awarded a total of two points, while those with fourfold symmetry receive three points.

6.2 Unique discrete building block representation

Consider a square with n patches per side, neglecting rotational symmetry. Also, assume that the patches cannot be flipped upside down. One side of the polygon then has 2^n possible patch configurations, and given that the polygon has 4 sides, the total number of unique configurations (without considering rotations or flips) is 2^{4n}. If we consider rotations, however, there will be double-counting of many of the configurations. For example, without rotation, a 4-sided system with 1 patch per side has $2^{4n} = 16$ configurations. With rotation, there is double-counting as shown in the figure below:

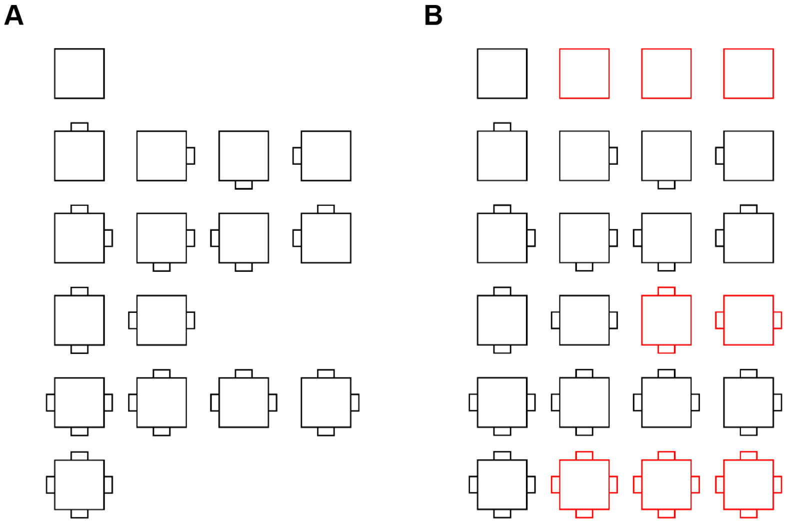

Figure 5: Illustration of the rotational symmetry of DON configurations. (A) All possible configurations for a 4-sided square. (B) Configurations (black) and the "missing parts" (red).

In fact, there are only 6 unique configurations when we consider rotations, not 16. A simple way to count all of the configurations is to "fill in" the missing parts of the figure above so that we can divide by the number of sides of the polygon to obtain the number of unique configurations, including rotation (Fig. 5B).

For square, there are two types of symmetry that will complete the figure above. The first is fourfold rotational symmetry, which appears in the first and last rows of the figure above. The second is twofold rotational symmetry, which appears in the fourth row of the figure above. For **fourfold** species, there are 3 members that need to be added to the group for every 1 symmetric species. For **twofold** species, there is 1 member that should be added for every 1 symmetric species.

To count the total number of unique members while considering rotation, we can then create a simple algorithm:

- Count the total number of members when we do not consider rotations, e.g. 2^{4n}.
- Add 3 to this count for every member that has order fourfold rotational symmetry.
- Add 1 to this count for every member that has order twofold rotational symmetry.
- Divide this count by the number of sides (4) to remove all rotations.

Counting the number of members with fourfold rotational symmetry is fairly simple since all sides of the square must have the same patch configuration. So, the number of these members is equal to the number of possible patch configurations, or 2^n. For twofold rotational symmetry, we must specify that there are two different patch configurations on the square. (They must, of course, be arranged in an alternating pattern.) This number is equal to $(2^n)(2^n - 1)$, with $(2^n - 1)$ being used to indicate that the second patch is different from the first. Note that this setup considers a trivial configuration without any patch, so a subtraction of 1 is needed. Completing the algorithm, then, leaves us with the following equation:

$$N_{\text{unique}}(n) = \frac{2^{4n} + 3(2^n) + (2^n)(2^n - 1)}{4} - 1 \tag{2}$$

Based on this equation, we obtained the unique patch locations of a 3×3 square $N_{\text{unique}}(3) = 1,043$.

6.3 Interaction settings in the molecular dynamics simulation

We treated the hydrophobic attraction between patches with a simple attractive shift-correct LJ potential U_{LJ} given by

$$U_{\text{LJ}}(r; \sigma, \varepsilon, r_c) = \begin{cases} 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} - \left(\frac{\sigma}{r_c}\right)^{12} + \left(\frac{\sigma}{r_c}\right)^{6} \right] & r < r_c \\ 0 & r \geq r_c \end{cases} \tag{3}$$

Imposing a standard cutoff distance of $r_c = 2.5\sigma$. We chose $\varepsilon = 1.0\varepsilon/k_{\text{B}}$, which can be parameterized further if experimental data is available, but it should only affect the transition temperature of the assemblies and not its final configuration.

For the rest of the interparticle interactions presented in the system (DNA origami-patch, DNA origami - DNA origami), we used repulsive Weeks-Chandler-Andersen (WCA) potential U_{WCA} given by

$$U_{\text{WCA}} = \begin{cases} 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] + \varepsilon & r < 2^{1/6}\sigma \\ 0 & r \geq 2^{1/6}\sigma \end{cases} \tag{4}$$

where $\varepsilon = 1.0\varepsilon/k_{\text{B}}$.

6.4 Dimensionality reduction on the structural properties space

To capture the high dimensional structural properties space and infer the phase regions of interest, we reduced the dimensionality of the descriptors by using either (i) simple linear reduction scheme like PCA (ii) advanced, non-linear dimensionality reduction technique that consider neighborhood and/or topology like t-SNE, UMAP, PacMAP, and TriMAP [32, 33, 34]. We highlighted the learned space based on the ensemble averaged of the absolute value of the bond orientational order $< |\psi_k| >$ with respect to various symmetries ($k = 2, 4, 6$) to understand the resulted plot. The results indicated that for all dimensionality reduction methods we tested here are able to separate designs with distinct structural properties (Fig. 6).

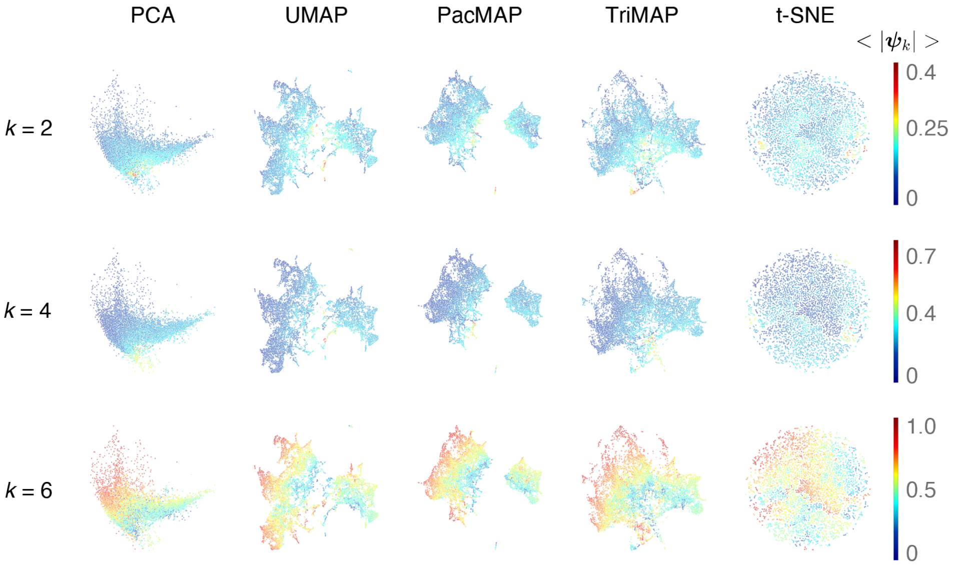

Figure 6: Dimensionality reduction on the structural properties space.

6.5 Detailed architecture of neural network

Below is the network hyper-parameter values for our best model with all three physical descriptors:

Parameter	Value
Input dimension (geometry)	13
Output dimension (physical descriptor)	167
Hidden layers	4
Neurons per hidden layers	1000
Batch size	128
Weight_decay (L2 norm strength)	0
Learning rate	1e-3
Learning rate decay schedule	0.8x when plateau at 10 epoch
Epochs	300
Optimizer	Adam

6.6 Symmetry-based data augmentation

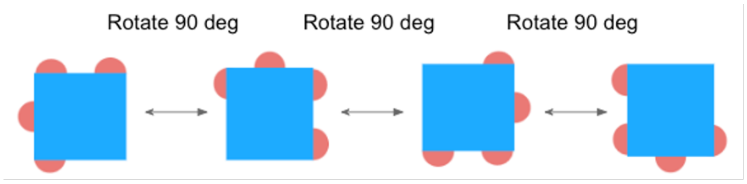

Figure 7: 1 Illustration of the rotational symmetry of the DON. the designs here are essentially the same physical configuration, but they are different in our parameterization.

6.7 Descriptor variance plot

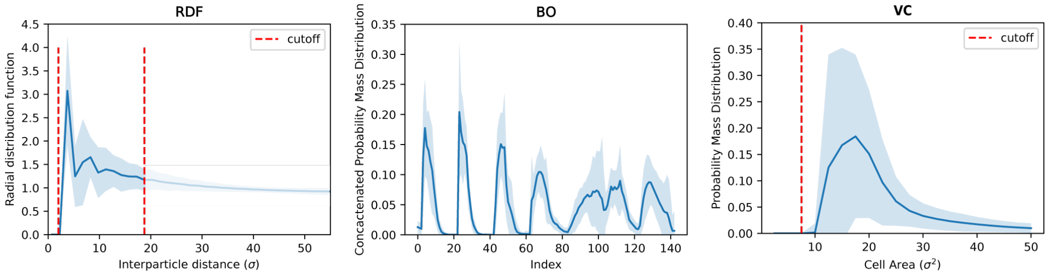

Figure 8: Variance of descriptors estimated across the entire dataset. For RDF (left), only parts of the RDF are used, ranging from 2 to 18.75σ (bounded by the red dash line), as the rest of the RDF not being informative. For BO (middle), the distribution within each symmetry is normalized. In VC (right), zero values are excluded (effectively, cell area smaller than $5\sigma^2$).