The Journal of Chemical Physics

Gaussian representation for image recognition and reinforcement learning of atomistic structure

Cite as: J. Chem. Phys. **153**, 044107 (2020); https://doi.org/10.1063/5.0015571
Submitted: 29 May 2020 . Accepted: 03 July 2020 . Published Online: 24 July 2020

Mads-Peter V. Christiansen, Henrik Lund Mortensen, Søren Ager Meldgaard, and Bjørk Hammer

AIP Publishing

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Gaussian representation for image recognition and reinforcement learning of atomistic structure

Cite as: J. Chem. Phys. 153, 044107 (2020); doi: 10.1063/5.0015571
Submitted: 29 May 2020 • Accepted: 3 July 2020 •
Published Online: 24 July 2020

Mads-Peter V. Christiansen, Henrik Lund Mortensen, Søren Ager Meldgaard, and Bjørk Hammer[a)]

AFFILIATIONS

Department of Physics and Astronomy, Aarhus University, DK-8000 Aarhus C, Denmark

Note: This paper is part of the JCP Special Topic on Machine Learning Meets Chemical Physics.
[a)]**Author to whom correspondence should be addressed:** hammer@phys.au.dk

ABSTRACT

The success of applying machine learning to speed up structure search and improve property prediction in computational chemical physics depends critically on the representation chosen for the atomistic structure. In this work, we investigate how different image representations of two planar atomistic structures (ideal graphene and graphene with a grain boundary region) influence the ability of a reinforcement learning algorithm [the Atomistic Structure Learning Algorithm (ASLA)] to identify the structures from no prior knowledge while interacting with an electronic structure program. Compared to a one-hot encoding, we find a radial Gaussian broadening of the atomic position to be beneficial for the reinforcement learning process, which may even identify the Gaussians with the most favorable broadening hyperparameters during the structural search. Providing further image representations with angular information inspired by the smooth overlap of atomic positions method, however, is not found to cause further speedup of ASLA.

I. INTRODUCTION

Machine learning (ML) has become a widely used tool in the quantum chemistry and computational materials science communities. Some of the most prominent applications of machine learning techniques so far have been in fitting potential energy expressions with unmatched accuracy to large databases of first-principles stabilities of molecular structures[1–4] and in establishing atomistic force fields by fitting against diverse structural samples and molecular dynamics trajectories for both molecular and solid state materials.[5,6]

The fitting of machine learning models, whether energy expressions or force fields, has been done, e.g., using kernel methods as in Gaussian process regression[7,8] or using artificial neural networks.[9,10] Some recent reviews of the rapidly moving field are given in Refs. 11 and 12. Examples of applications involving kernel models include studies of structural properties of liquid and amorphous phases of carbon and silicon,[13,14] melting transitions of metallic, semi-conducting, and oxidic materials,[15,16] and prediction of energy–pressure relationships of super hard tungsten nitride bulk materials.[17] Examples of applications based on artificial neural networks include the studies of the ground state structure of solid boron[18] and metallic nanoparticles,[19–21] the phonon dispersion relations and melting temperatures of bulk semiconductors and metals,[22] and the self-diffusion properties of metal surfaces.[23,24]

A common theme for all the mentioned machine learning models, whether kernel or neural network based, is that they need to represent the constituent atoms by the so-called *features* that capture the chemically important aspects for the given regression or classification task. The field has been driven by the proposal of efficient hand engineered features, starting from a series of radial and angular atomic symmetry functions as in the Behler–Parrinello approach[1] and so far reaching highly complex representations such as the smooth overlap of atomic positions (SOAP),[25] Faber–Christensen–Huang–Lilienfeld (FCHL),[26] and graph-based representations where nodes correspond to atoms and edges correspond to bonds.[27,28] Identifying the ideal feature for a new task can be a cumbersome process, which led to the development of neural network architectures designed to learn the representation from data. Such architectures include SchNet,[29] which uses continuous convolutions to iteratively improve an atomic representation.

When introduced in connection with global structure optimization, machine learning models may be constructed as accurate

surrogate energy landscapes that allow for most of the computationally demanding local relaxation steps to be performed at the model level.[30] Once a model relaxed candidate structure is obtained, a single or a few single-point energy evaluations at the first-principles level may be conducted. Subsequently, the machine learning model may be updated, and hence, a more and more reliable model may be established in an active learning setting while the global structure optimization is completed.[19,21,31–34] Other schemes for machine learning enhanced global structure optimization have, however, also been proposed in which the ML models are intentionally not some surrogate energy landscapes. Rather, the ML models may provide local energy information,[35,36] serve to provide uncertainty measures to balance exploration vs exploitation,[37–39] or act to remove energy barriers by adding extra dimensions[40] or making energy landscapes simpler and more convex.[41,42]

Recently, an entirely different approach to global structure optimization, namely, utilizing reinforcement learning, was proposed.[43,44] In our work,[43] we utilize image recognition techniques to have a machine learning agent learn by itself how to build atomistic structures. The method Atomistic Structure Learning Algorithm (ASLA) revolves around a convolutional neural network (CNN) based agent, which directs the construction of new candidate structures while actively learning about the stability of the built structures by interacting with a first-principles total energy evaluator, such as a density functional theory (DFT) program. Compared to traditional global optimization methods, such as an evolutionary algorithm, that often rely on local relaxations that can require many force evaluations, the ASLA requires only a single energy calculation per candidate structure (see Ref. 45 for a comparison of the computational demands). Furthermore, problem specific mutations are not required as the algorithm learns to build high quality candidates with the information stored in the CNN, rather than iteratively generating them through chance. An additional strength of the method is the ability to transfer knowledge from one problem to another, as showcased in the original paper. In our first application of ASLA, we employed a one-hot encoding of the atomic positions, meaning that the neural network agent takes as input discretized images of space in which pixels are 1 wherever atoms are present and 0 elsewhere. The purpose of the present work is to investigate how the input representation of atomistic structure may be augmented utilizing ideas from the general field of machine learning in chemical physics and to probe to what degree an improved representation may speed up the function of ASLA in determining global minimum energy structures.

CNNs have been used for drug-discovery applications, and different representations have been suggested in this field. Wallach *et al.* presented a CNN for this task based on protein–ligand descriptors of differing complexity evaluated at each grid point.[46] Torng and Altman used a CNN with a one-hot representation followed by a Gaussian filter for amino-acid environment classification,[47] whereas Ragoza *et al.* used a piecewise Gaussian and quadratic function as the representation for a CNN for protein–ligand scoring.[48] In both of the latter works, the length scale of the Gaussian was chosen based on the van der Waals radius of the atom. Kuzminykh *et al.* showed that a wave representation, with the filter being a Gaussian multiplied by a sine function, leads to a lower reconstruction loss of an autoencoder when compared to a pure Gaussian representation.[49] The above works share the common feature that they work in a supervised learning setting, that is, the neural network is trained based on a predetermined dataset and the important performance measure is accuracy on an unseen validation set. This is very different from the ASLA setting, wherein the algorithm produces its own training data as it progresses. An unfortunate choice of representation can lead to the algorithm stagnating whereby it becomes unable to generate training examples that can further the search. A more suitable choice of representation would then lead to a significant decrease in the amount of computational effort required to find the global minimum structure, as will be shown.

This paper is organized as follows: Sections II and III introduce ASLA and present the impact of replacing the one-hot encoding with a radial on-site Gaussian broadening, respectively. Section IV extends the representation to one with a background bias. Section V probes the possibilities of automatically finding an appropriate Gaussian representation by expanding the kernel in a basis of on-site Gaussians. Section VI extends this basis expansion to displaced Gaussians so as to generate a more general radial kernel. In Sec. VII, we investigate the effect of representing atoms in a SOAP-like manner with displaced Gaussians and angular terms. Finally, in Secs. VIII and IX, we discuss the usability of the developed method for multi-component systems investigated with a full DFT description and conclude this paper, respectively.

II. ASLA

The general layout of the atomistic structure learning algorithm (ASLA) is shown in Fig. 1(a). The ASLA contains a reinforcement loop of episodes, each of which has three main phases: a building phase, an evaluation phase, and a training phase. The *building phase* involves an agent in the form of a convolutional neural network that produces a structural candidate, the *evaluation phase* is a sole single-point (i.e., unrelaxed) total energy calculation, e.g., performed within DFT, and the *training phase* is a back-propagation step improving the agent. The building phase is further detailed in Figs. 1(b) and 1(c). The new structures are built atom by atom in a discretized space, possibly starting from a template, where some atoms are already placed. Space may be a 2D layer or a 3D matrix, and different atom types are treated in separate input layers or matrices. For each atom to be added, an iteration is taken in which the current, incomplete structure is input to the agent, which outputs a Q-value map. This map details the agent's current expectation of how stable a final atomistic structure can become if the next atom is placed at any given pixel in the map. Maximum Q-values of 1 indicate that the most stable structure is expected, while smaller Q-values down to -1 predict less stable final structures. In most iterations, a greedy action is taken, meaning that the atom type and position are chosen according to where the agent outputs the maximum Q-value. However, to provide some exploration, the atom type and position are sometimes chosen entirely randomly or randomly among a fraction of the highest Q-values. The reinforcement learning comes about since the agent keeps building according to its expectation, while at the same time, it keeps being confronted with the reality once the structure is finalized and its stability calculated in the evaluation phase. This means that every predicted Q-value for atoms actually placed during the build iterations can be associated with a target Q-value depending on the energy calculated in the evaluation phase.

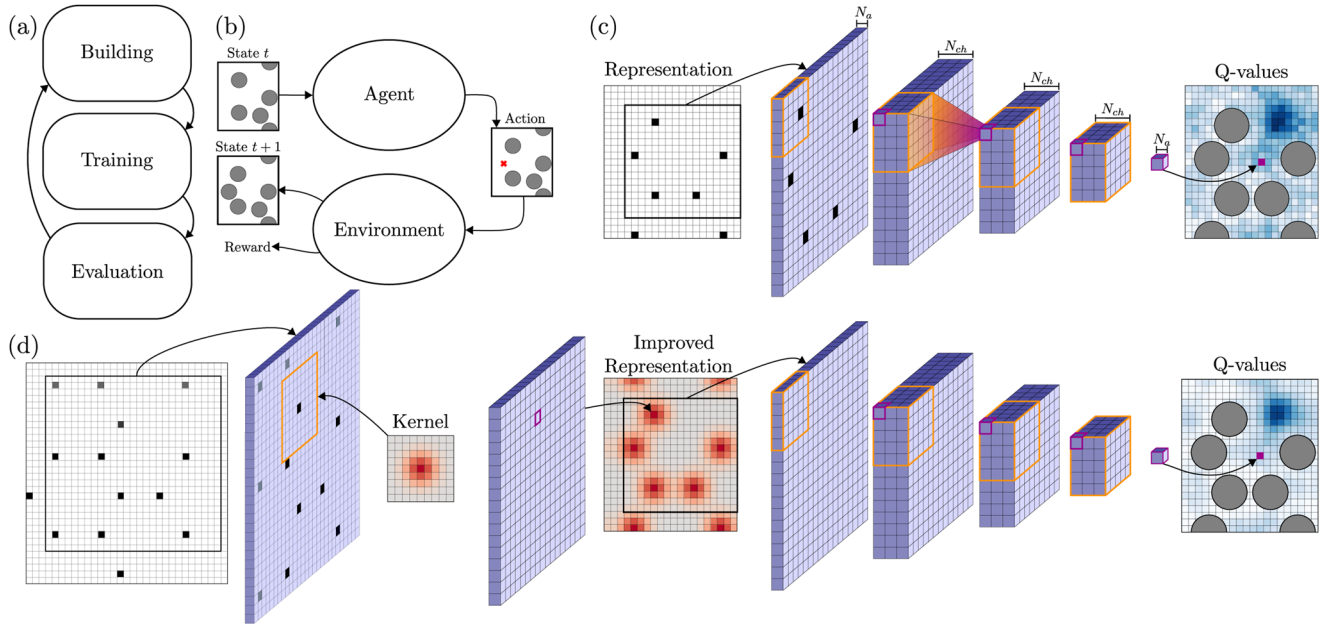

FIG. 1. (a) General layout of ASLA. (b) The cyclic build action, where the agent provides Q-values that determine position and type of next atom to place. (c) The action of the agent following ASLA as in Ref. 43: The one-hot representation of atoms is input directly to the deep convolutional neural network (CNN). (d) This work: The one-hot representation of atoms is converted using a convolutional layer before being fed into the standard ASLA deep CNN.

This is what is done during the training phase, which involves experience replay, symmetrization augmentation, and back-propagation. The results presented in this work were conducted using a CNN with four convolutional filters, with the filters of the hidden layers having 10 kernels of size 15 × 15, resulting in a total of 49 531 trainable weights. For further details, see Refs. 43, 45, and 50.

III. GAUSSIAN REPRESENTATION

To probe the effect of how atoms are represented to the agent in ASLA, we start by modifying the input, as shown in Fig. 1(d). Now, the original one-hot encoding is modified by applying a single hard-coded convolution kernel that runs over the entire input image using appropriate periodic or zero-padding boundary conditions depending on the problem at hand. This "representation kernel" has dimensions 25 × 25, zero bias, and weights according to a Gaussian,

$$K_\lambda(r) = \mathrm{e}^{-\frac{1}{2}r^2/\lambda^2}, \tag{1}$$

where r is the distance from the mid-point of the convolution kernel, i.e., pixel (13,13), and the hyperparameter λ controls the width of the Gaussian. The large size of the representation kernel ensures that the kernel is approximately zero at the edge even for large λ without having to introduce a cutoff function. Note how K_λ conveniently collapses to a delta-function and hence a one-hot encoding in the limit of small values of the hyperparameter λ, $\lambda \to 0$.

As a test system, the ASLA is setup to build a pristine graphene sheet by placing 23 C atoms in a periodic cell, where initially, one C atom is present. This system, previously considered in Ref. 43, has a grid of 41 × 36 pixels^2 with a side length of $\Delta \approx 0.20$ Å. Energy calculations are performed with density functional tight-binding (DFTB) using the DFTB+ program,[51] as the algorithm needs to be restarted a large number of times to obtain reliable statistics. The trade-off in accuracy of using DFTB compared to DFT is not of importance as we are not interested in specific chemical properties, rather in the algorithm's performance as a global search method. In fact, the global search problem is actually more difficult in the DFTB landscape, likely due to minima not present in DFT.

In order to study the effect of the Gaussian representation and the importance of the hyperparameter λ, we have determined the success curves of ASLA as a function of λ. Two such curves are presented in Fig. 2(a). They are each based on a large number of restarted runs with new random initialization of the weights in the CNN agent. The success curves then record as a function of completed reinforcement episodes the share of restarts that have found the global minimum (GM) energy structure. As indicated with the blue dashed line, the median number of episodes, $n_{50\%}$, required for runs to successfully identify the GM is ≈750 when using the one-hot encoding. Resorting, however, to the Gaussian representation, the median number of episodes required reduces significantly and becomes less than 500. The Gaussian used in Fig. 2(a) has $\lambda = 1.25\Delta \approx 0.25$ Å, which appears to be the optimal value, as evidenced by Fig. 2(b), showing the $n_{50\%}$ vs λ.

The reason for the faster identification of the global minimum energy structure using the Gaussian representation may be sought

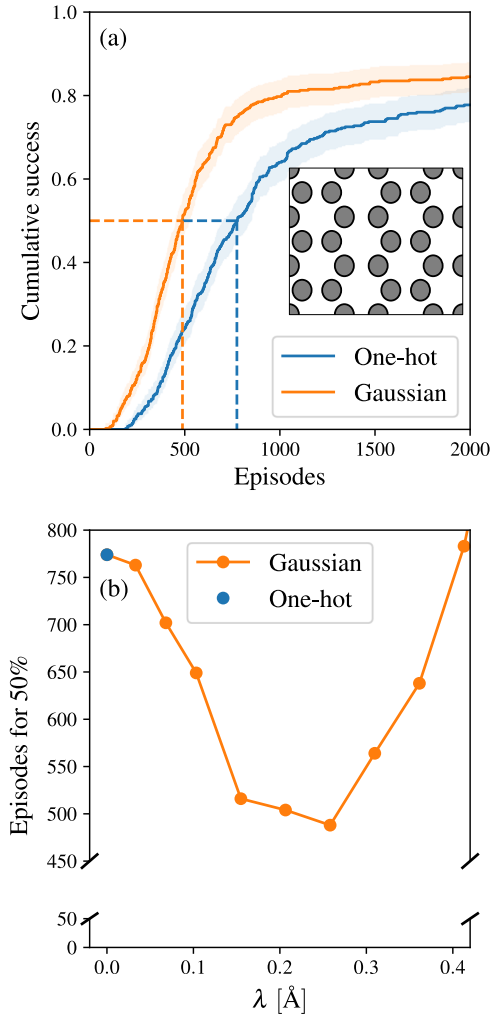

FIG. 2. (a) Success curves for one-hot and Gaussian representation with $\lambda = 1.25\Delta \approx 0.25$ Å; dashed lines mark the number of episodes required to reach 50% success rate. (b) Episodes required for 50% success rate as a function of λ.

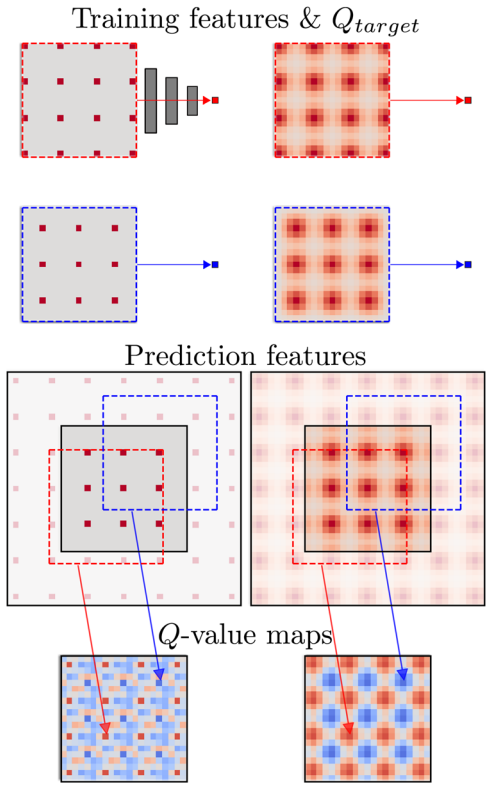

FIG. 3. Use of a one-hot representation (left hand side) vs Gaussian representation (right hand side). The agents are trained in a supervised training setting on two pieces of data only (top part), and the depicted Q-values should be reproduced. For the top left feature, this is illustrated by a schematic network and an arrow; the arrows for the other three features should be understood as the same operation of feeding the feature through the network. Once applied over the entire image of an extended structure (middle part), they provide predicted Q-value maps (lower part). The extended structures are shown dimmed outside a unit cell, highlighting the use of periodic padding. The dashed squares over the extended images identify inputs corresponding to those present in the training data, and the arrows indicate the learned Q-values that are in accordance with the training data. Using one-hot encoding, other predictions in the Q-value map appear noisy, while using a Gaussian representation, the agent appears to generalize better.

in an improved ability of the agent to generalize from such an input. To substantiate this, we present in Fig. 3 examples of two agents that have been trained in a supervised learning setting, either on a one-hot representation or on a Gaussian representation. Only the two pieces of training data shown in Fig. 3 (top) are provided. Subsequently, the trained agents are presented with the structures shown in Fig. 3 (middle), leading to the Q-value maps of Fig. 3 (bottom). It is seen that both agents have been trained sufficiently to reproduce the target Q-values completely whenever the kernels pass over regions [dashed boxes in Fig. 3 (middle)] of the input image that overlap with the training data. However, the agent trained and tested on the one-hot encoded input appears to produce random noise for elsewhere. Contrary to this, the agent trained and tested on the Gaussian represented input does provide a smoothened Q-value map and hence predicts what appears to be meaningful Q-values throughout the image. The tendency becomes particularly apparent in the present example due to the translational symmetry of the structure considered. However, the effect will also be present in less symmetric environments and with other training data. It reflects that by smearing the atomic position on the input, more environments will be conceived by the neural network as resembling the training data and will consequently lead to the predicted Q-values influenced by the training data. The degree to which such favorable generalization from training data to test data appears depends on the λ hyperparameter. Since the Q-values essentially represent rescaled energies, the optimal choice of λ thus reflects, to some degree, the natural length scale for energy variations, i.e., some measure of how much each atomic position can be changed while still expecting a similar stability of the structure. In this light, the reported $\lambda = 0.25$ Å appears sensible.

IV. BIAS

The results presented up until now were obtained with zero bias in the convolution that converts the input from the one-hot representation to a Gaussian representation. However, including a single bias as an extra degree of freedom in this conversion would lead to improved performance. This is illustrated in Fig. 4 where statistics is shown for ASLA when run with either a fixed negative (−0.1) or a fixed positive (+0.1) bias. Using the negative bias is beneficial to the method, improving $n_{50\%}$ from ≈500 with no bias to ≈400. On the other hand, using a positive bias is seen to be detrimental as $n_{50\%}$ increases to ≈700. Note that these results are obtained without optimizing the biases, while λ is reoptimized in the presence of a bias.

To explain the favorable effect of the negative bias, it is instructive to consider the value of pixels far from any atoms. In the presence of the negative bias, such pixels become negative and the kernels in the ASLA network can thus operate on these regions, e.g., one could imagine that the ASLA CNN could amplify these negative values to easily learn that Q-values of −1 should be attributed to these regions. At first thought, the same argument should hold for a positive bias, but the problem arises that it becomes difficult for the network to distinguish between pixels that are close to an atom and hence positive due to the tail of the Gaussian or are far from an atom and therefore positive due to the bias. In fact, if instead the Gaussian kernel is chosen to have a negative amplitude, the effect of the sign of the bias becomes reversed. With a bias of zero, the kernels of the ASLA CNN can only operate on regions close to atoms, and these degrees of freedom can therefore not be used efficiently.

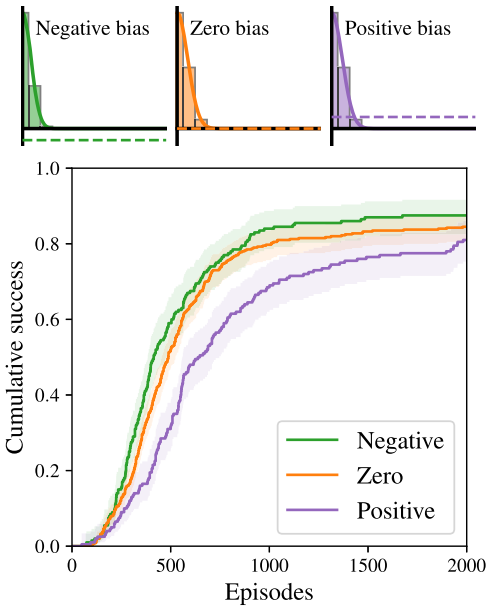

FIG. 4. Top panel illustrates the optimal Gaussian kernel with negative, zero, and positive biases. The same kernel is found for positive and zero biases, whereas a slightly more shallow kernel is found for negative bias. In the bottom panel, the success statistics for the runs are shown.

V. AUTOMATED GAUSSIAN REPRESENTATION

The kernels for the conversion of the one-hot representation to the Gaussian representation used in Secs. III and IV have been hard-coded based on the chosen values of λ and bias. With proper choices, the ASLA exhibited improved performance. However, it is also clear that if either λ or the bias is chosen wrongly, the improvement can become insignificant or, in the worst case, degrade. Scanning for the optimal values of these hyperparameters is computationally expensive and would negate the gain in performance. We hence propose in this section to follow a different strategy, namely, to consider the representation network and ASLA's deep CNN as *one combined* network, whereby the representational hyperparameters may be optimized as an integral part of the backpropagation, when the loss function (the mean-squared error between Q-values and the rewards calculated based on structural energies) is minimized.

The simplest approach would be to make λ trainable and to write the representation kernel as a function of it. However, doing so, we identified a strong dependence of the resulting optimized value of λ on the initialization of λ. The issue was circumvented by expanding the kernel in a basis of Gaussian kernels of predetermined widths,

$$K(r) = \sum_i c_i e^{-\frac{1}{2}r^2/\lambda_i^2}, \tag{2}$$

where the expansion coefficients, c_i, become the hyperparameters trained according to the loss function. The set of λ's is chosen in terms of the grid spacing, as $\lambda_n = \{\frac{1}{3}\Delta, \frac{2}{3}\Delta, \Delta, \frac{4}{3}\Delta, \frac{5}{3}\Delta, 2\Delta, \frac{7}{3}\Delta, \frac{8}{3}\Delta, 3\Delta\}$. Some of these "basis kernels" are shown schematically in Fig. 5 alongside the remainder of the network to clarify the architecture. In order to improve the training of the expansion coefficients, the representation resulting from the convolution between the basis expanded kernel and the one-hot representation is *batch normalized*, a technique commonly used in neural networks.[52] Since batch normalization contains a learnable bias, this effectively provides the bias, which was identified in Sec. IV to be beneficial to the ASLA's success rate. The basis coefficients are initialized randomly from a uniform distribution in the range [0, 0.5].

The procedure just described is named the *Automated Gaussian Representation (AGR) method*, and ASLA's performance using it is demonstrated in Fig. 6(a). It is seen that the success curve of ASLA with the AGR method exceeds that of ASLA with the one-hot representation and that it yields a comparable performance to that of ASLA when the Gaussian representation of optimal, predetermined λ is employed. This is a striking result showing that learning the representation on-the-fly represents only a minor further complication of the overall problem.

Figure 6(c) presents a histogram of the expansion coefficients that were found after 1000 episodes. The coefficients, which have been averaged over 250 restarts, show a rather broad distribution over the kernels of the various allowed λ values. This is perhaps surprising given that Fig. 2(b) showed a rather narrow range of well performing λ values for *predetermined* kernels. The analogy may, however, be recovered by adding L_1 regularization to the loss function, thereby penalizing sizable expansion coefficients on less useful kernels. Specifically, we add

$$L_1 = \beta_1 \sum_i |c_i| \tag{3}$$

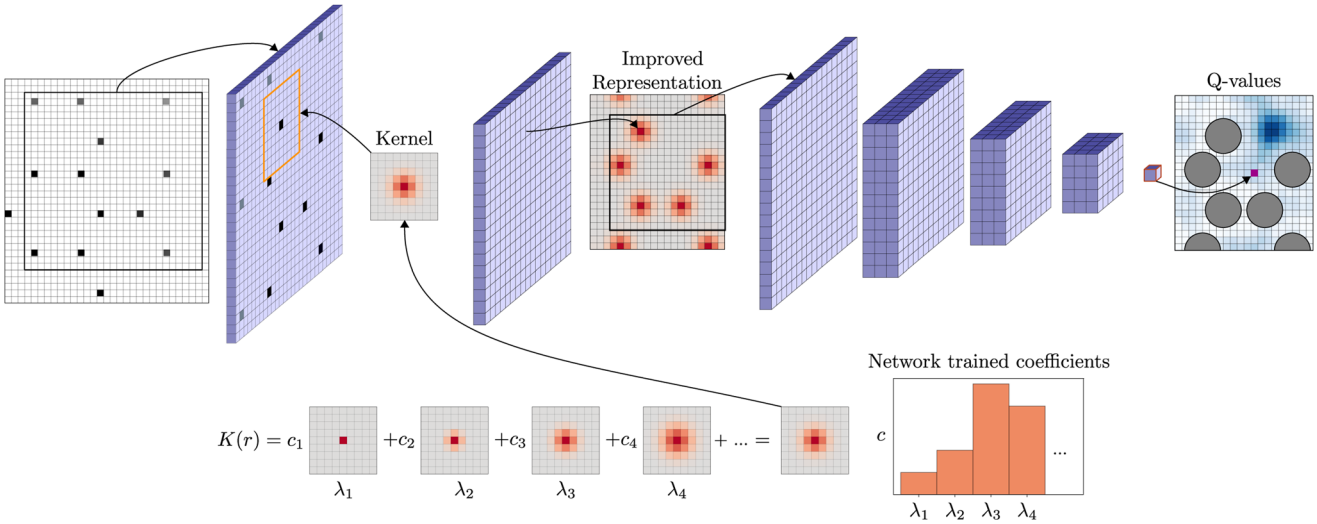

FIG. 5. Schematic of the kernel basis expansion method. The ASLA network is prepended with additional weights that are interpreted as the expansion coefficients of a predetermined set of basis kernels.

to the loss function, where β_1 is a weighting factor that determines the importance of the L_1 term. The success curves of ASLA shown in Fig. 6(b) appear largely unaffected by the addition of the L_1 regularization, but the distributions of coefficients now become more narrowly peaked around the kernels of optimal λ ($\sim\lambda_4$) values, as evidenced by Figs. 6(d) and 6(e).

To further test the AGR method, we turn to a significantly harder problem: the atomic structure inside a grain boundary in a 2D graphene sheet. The geometric setup is illustrated in the inset of Fig. 7. The template provides the C atoms at the positions of the light gray disks setting up the edges of two carbon sheets tilted 13.17° with respect to each other. The ASLA is tasked with

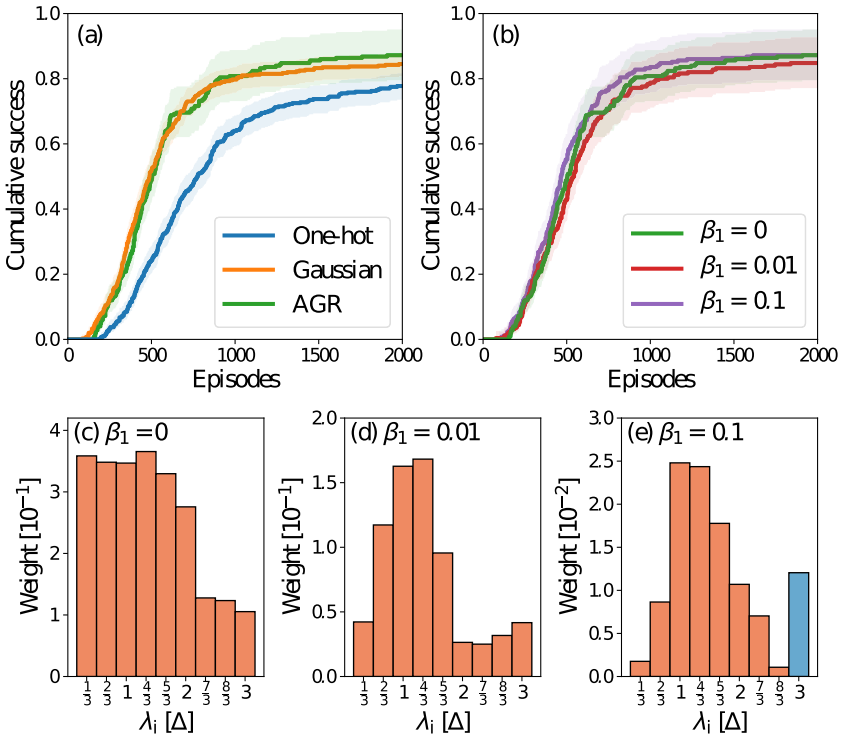

FIG. 6. Use of the automated Gaussian representation (AGR) method for solving the problem of placing 23 C atoms as graphene on a periodic template with 1 C atom present. (a) Comparison of success curves to one-hot representation and the best single width Gaussian from Fig. 2(b). Comparison of success with the AGR method using varying degrees of L_1 regularization. [(c)–(e)] Histograms of mean expansion coefficients for β_1 = 0, 0.01, and 0.1, respectively. Orange bars indicate positive values, and the blue bar indicates a negative value.

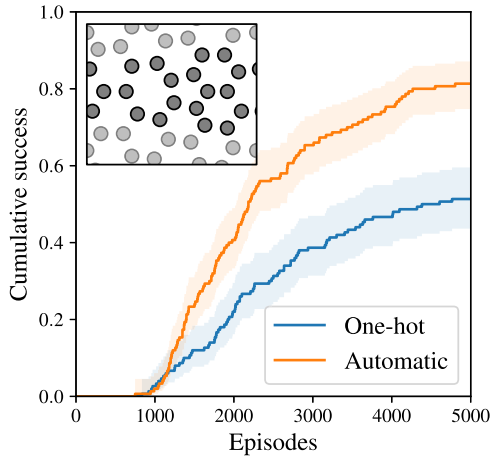

FIG. 7. Success curves for one-hot and AGR method on grain boundary between two angled graphene sheets shown in inset.

placing 20 C atoms in the void between the two edges. Eventually, the ASLA will find the global minimum energy structure supported by the grid, where the 20 C atoms assume the positions of the dark gray disks.

The success curves for solving this grain boundary structure are given in Fig. 7. Using the one-hot encoding, the ASLA requires 4400 episodes on average to solve the problem correctly with 50% likelihood. However, when using the AGR method, the ASLA requires only about 2200 episodes to reach the same fidelity and attains more than 80% success after 5000 episodes. The AGR thus proves highly successful when used in solving this more difficult structural optimization problem.

VI. DISPLACED GAUSSIAN BASIS

The basis used in Sec. V was a set of atom-centered Gaussians with different width parameters, λ_i. In this section, we shall instead probe the use of *displaced* radially Gaussians while settling on the use of a single value of λ. The kernel expansion now looks like

$$K(r) = \sum_i c_i e^{-\frac{1}{2}(r-r_i)^2/\lambda^2}, \tag{4}$$

where the c_i's are again expansion coefficients but now acting on basis functions, where an offset r_i has been introduced. For a 25×25 kernel, as has been used thus far, with a basis set of seven Gaussians with $r_i = 2i\Delta$ with i going between 0 and 6, where λ then controls the overlap between neighboring Gaussians, here, $\lambda = \Delta$ is chosen. Selected basis kernels are displayed in Figs. 8(a)–8(d). The initialization of the expansion coefficients decides the initial shape of the kernel. Three initialization strategies were explored:

(i) random uniform initialization of the expansion coefficients between −1 and 1,
(ii) random initialization of the expansion coefficients as in (i) but modulated by a linearly decreasing envelope function that decays from 1 for $i = 0$ to 0.1 for $i = 6$, and
(iii) initialization to mimic a near-optimal kernel composed of one Gaussian, as identified in Fig. 2(b), i.e., having $c_0 = 1$ and the remaining trainable c_i's initially set to 0.

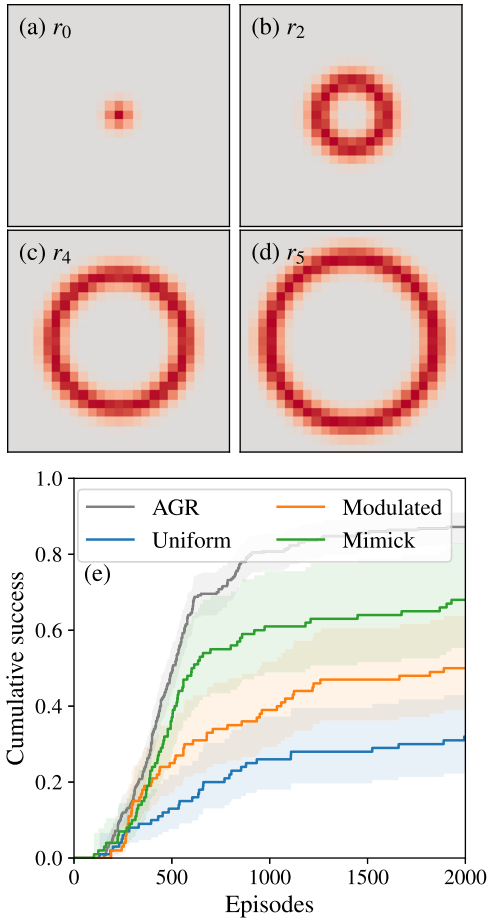

FIG. 8. [(a)–(d)] Selected displaced Gaussian basis kernels. (e) Success curves for the displaced Gaussian basis representation with the three initialization strategies compared to the AGR method.

In all three cases, the expansion coefficients were trained as described for the AGR method but without L_1 regularization.

The ASLA success curves for solving the periodic graphene problem with 1+23 C atoms are given in Fig. 8(e). It is seen that the first two types of initialization provide unfavorable representations that significantly delay ASLA's finding of the global minimum energy structure. Only the final choice where knowledge of a known optimum representation is utilized is reasonably successful, yet inferior to the use of a single on-site Gaussian. We conclude that when selecting Gaussian type kernels for the representation of atomic structure as blurred images, the choice of a collection of on-site radial Gaussians of varying widths appears to be more robust than the selection of a collection of differently displaced radial Gaussians. The latter type of Gaussians will, however, be used in Sec. VII, where we further consider the representation of angular information.

VII. ANGULAR KERNELS

So far, only radial information has been incorporated in the representation, and ASLA's CNN network has to learn about bond angles in later layers. In this section, we will, however, be exploring if providing angular information immediately in the image representation is capable of speeding up the ASLA's learning rate.

Naively, providing angular information about every pixel in an image of an atomistic structure may sound like a formidable task. At any given empty pixel in an image, information about a bond angle may be of relevance if placing an atom at that pixel does lead to the formation of at least two bonds, whose mutual angle may be derived. Formulated in this way, the computational task does not lend itself to be solved with kernels in a CNN setup, as has been done for the radial information up until now. However, if we apply the same trick as done in the construction of the SOAP feature, the angular information can easily be derived from the application of convolutional kernels.

In SOAP, the power spectrum of the spherically harmonic expansion of Gaussian broadened atomic positions provides the required angular information. SOAP has the obvious advantage over standard protocols for obtaining bond angle information, such as the one found in the Behler–Parrinello approach, that it does not contain any double sum over neighboring atomic positions. It is the very same virtue that makes it useful in the present context.

In the present scope with the usage in a 2D setup with ASLA, it suffices to consider the angular information starting from the circular harmonic functions,

$$K_m(\theta) = \mathrm{e}^{im\theta} = \cos(m\theta) + i\sin(m\theta). \tag{5}$$

Figure 9 defines the angle θ and depicts the $K_m(\theta)$ coefficients for several m. These complex-valued coefficients can be used in combination with a radially displaced Gaussian kernel,

$$K_{\lambda,r_s}(r) = \mathrm{e}^{-\frac{1}{2}(r-r_s)^2/\lambda^2}, \tag{6}$$

to calculate a real-valued representation in the following way. First a general complex kernel is defined as

$$K_{\lambda,r_s,m}(r,\theta) = K_{\lambda,r_s}(r)K_m(\theta), \tag{7}$$

the point at which a real-valued representation can be calculated as

$$R^{\mathrm{out}}_{\lambda,r_s,m} = \left(\mathrm{Re}[K_{\lambda,r_s,m}(r,\theta)] * R^{\mathrm{in}}\right)^2 + \left(\mathrm{Im}[K_{\lambda,r_s,m}(r,\theta)] * R^{\mathrm{in}}\right)^2, \tag{8}$$

where $*$ should be understood as the convolution operator and R^{in} is the original one-hot encoded image that is being converted to an output channel, $R^{\mathrm{out}}_{\lambda,r_s,m}$. This allows the construction of representations using multiple stacked channels, with differing parameters λ, r_s, m, analogous to RGB images that use three channels, one for each color. The convolutions act as the inner products leading to the expansion coefficients of the smeared atomic densities in the circular harmonics representation, and the complex squaring provides the value of the power spectrum at the given m. The use of convolutions assures that the information is constructed for every pixel in the image in one go. A similar methodology was proposed by Ref. 53 for constructing rotationally invariant CNNs and expanded upon by Ref. 54 for 3D geometries, whereas Ref. 55 used a wavelet based rotationally invariant network for predicting molecular energies in a supervised setting.

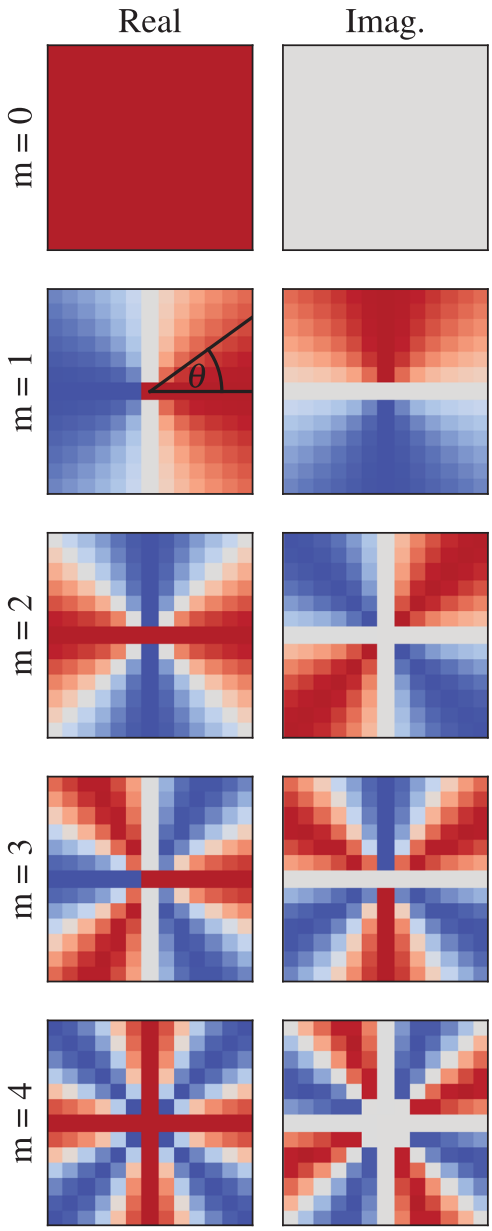

FIG. 9. Real and imaginary parts of circular harmonic coefficients of different angular orders m. The angle θ is calculated between the horizontal and the line going from the central pixel to the pixel of interest.

Figure 10 depicts how Eqs. (7) and (8) can be implemented in practice using convolutions and element-wise products that are available in most neural network software packages, such as

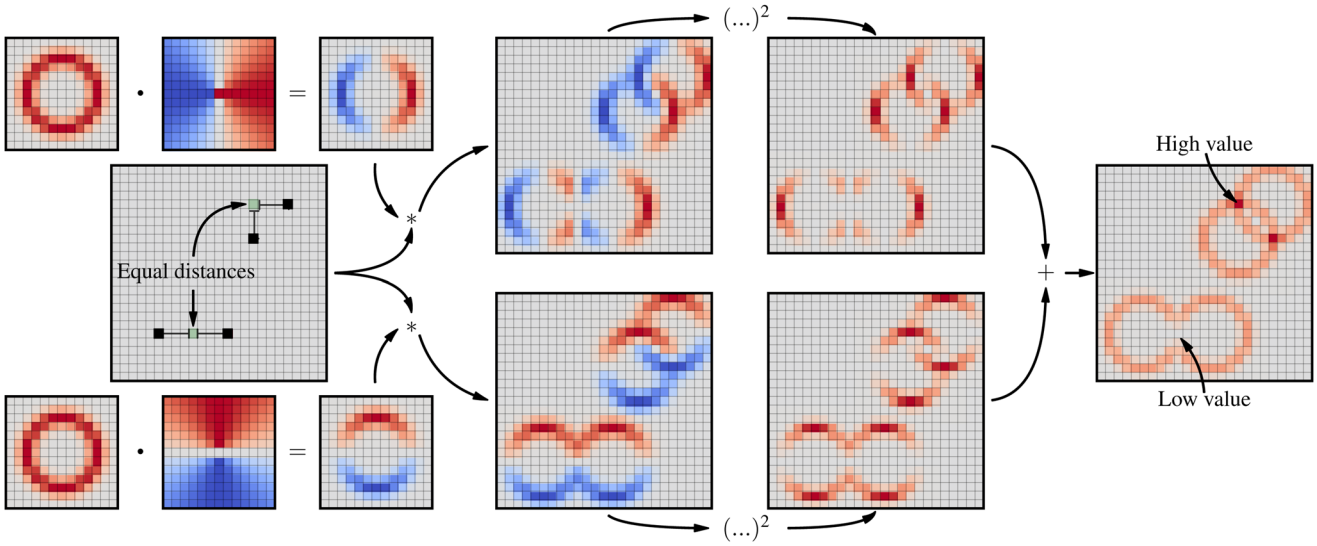

FIG. 10. Radial and angular kernels are separately multiplied to obtain the real and imaginary parts of a general equivariant kernel, which are then convolved with a one-hot representation, which is subsequently squared and summed leading to the final equivariant representation.

Tensorflow.[56] The figure uses a small λ and an r_s that is close to the ideal bond distance and probes for $m = 1$. In the input image, two different pixels are highlighted. Both pixels are placed at a distance r_s from two neighboring atoms, but placing an atom at one of the pixels would form three atoms in a linear configuration, while placing an atom at the other pixel would form three atoms in a 90° bent configuration. The $m = 1$ part of the power spectrum is capable of discriminating these two situations as evidenced by the output representation. The pixel that would lead to a linear configuration attains a low value, while the pixel that would lead to a 90° bent configuration ends up having a large value. The corresponding $m = 0$ channel (not shown) is purely radial and would not discriminate between the two pixels, but provide the same large value for both of them. Representing the atomistic structure with a series of different m (in principle, also λ and r_s) values thus seemingly facilitates the learning that must take place in ASLA's deep CNN.

To probe the usefulness of presenting the ASLA with radial and angular information in such stacked input representations, we

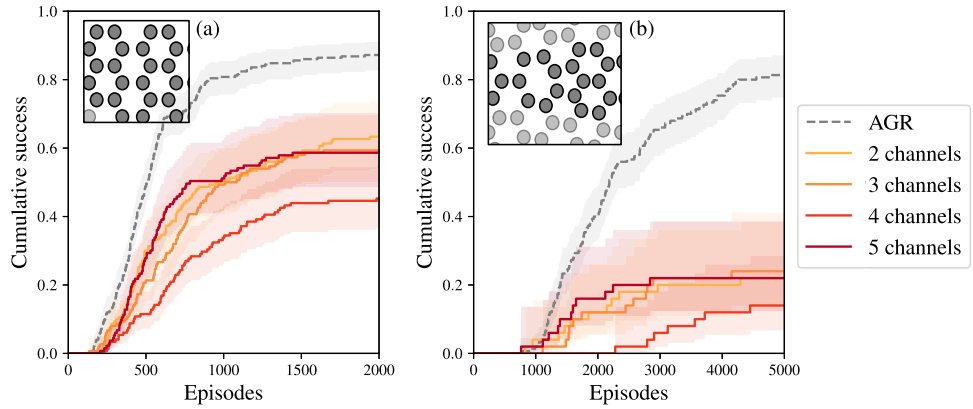

FIG. 11. Tests of multi-channel representations for (a) the 1 + 23 C pristine graphene problem and (b) the 20 C graphene grain boundary problem. The two-channel representation consists of an on-site Gaussian with $\lambda = 1.5\Delta$ and a Gaussian with $r_s = 1.4$ Å with $\lambda = \Delta$, both with $m = 0$. [Note that while $\lambda = 1.5\Delta$ was not the optimal value for a single on-site Gaussian found in Fig. 2(b), it is the optimal value for a similar search if batch-normalization is utilized, as it is here.] The three-channel representation uses the same two channels and an additional $m = 1$ channel with $r_s = 1.4$ Å and $\lambda = \Delta$. The four- and five-channel representations are constructed in the same way by stacking with additional channels having $m = 2$ and $m = 3$, respectively, again with $r_s = 1.4$ Å and $\lambda = \Delta$. Generally, the more the input channels are present, the harder it is for the ASLA to solve the problem. However, when the 120° channel is added ($m = 3$), a slight improvement is observed. The performance of the AGR method is shown with the dashed curve for reference.

conducted a number of structural searches with ASLA for the perfect 1+23 C graphene problem and the 20 C grain boundary problem. In none of the cases, however, did we find that the ASLA benefited from having angular information provided over just having radial information. The details of these results are given in Fig. 11.

Starting with the 1+23 C graphene problem in Fig. 11(a), it is seen that the best performance is obtained with the purely radial AGR method described above. Adding r_s = 1.4 Å channels with $m = 0$, 1, and 2 progressively deteriorates the success curves, suggesting that the ASLA finds no useful information in these extra input channels and that the ASLA spends costly reinforcement episodes realizing this. Only when arriving at the m = 3 channel, a relative improvement of ASLA is seen. This is not surprising since the m = 3 part of the power spectrum is capable of identifying and discriminating between positions that would lead to the formation of 60° and 120° bond angles, which is precisely what is needed to learn to build honeycomb rather than hexagonal close packed structures. However, despite the improved performance for ASLA when presented with input channels of angular information up to m = 3, it does, according to the success curves in the figure, not identify the favorable information fast enough to outperform the situation when only fed with the radial input as with the AGR method.

The same scenario plays out for the harder 20 C graphene grain boundary problem, as seen from Fig. 11(b). Again, the AGR method shows a high success rate, which cannot be matched in runs where angular channels are provided as input. Again, some improvement is observed when the expansion in angular components include the m = 3 term, but not sufficient for the success curve to become even comparable to that of the AGR method. We are led to conclude that useful angular information may indeed be provided to ASLA via SOAP-inspired transformations of a one-hot encoded representation of the atomic structure. However, the ASLA, when provided with multiple input channels, must identify which channels are useful, which causes a delay in ASLAs learning, and which is not compensated by the presence of more concise information in some of these extra channels.

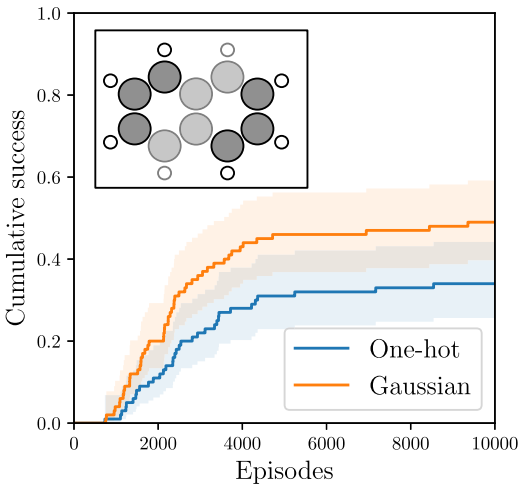

FIG. 12. Success curves for the organic system with one-hot and Gaussian representations. Inset: global minimum energy structure of naphthalene with the template atoms shown dimmed.

TABLE I. Timings for the three phases of an ASLA episode averaged over the 10 000 episodes of 15 restarts with 5 cores of a 2.1 GHz Intel Xeon Gold 6230 processor. Numbers in parentheses indicate the percentage of an entire episode for each phase. The additional time used during building and training caused by the extra convolution required for the Gaussian representation is negligible compared to the time of a DFT calculation.

Phase	One-hot (s)	Gaussian (s)
Building	0.81(5.6%)	0.95(6.4%)
Evaluation	10.39(72.3%)	10.24(69.0%)
Training	3.17(22.1%)	3.65(24.6%)

VIII. OUTLOOK: MULTIPLE SPECIES

Thus far, we have shown that for systems consisting of a single atomic species, the performance of ASLA improves given a better representation. In this section, we report on initial findings for multi-species systems, namely, on the ASLA tasked with building naphthalene. The performance of a one-hot representation for each atomic species is compared to a Gaussian representation for each species, but with the same λ of 1Δ with a 15 × 15 kernel, the resulting success curves are shown in Fig. 12. A significant performance increase is observed without fine-tuning λ for each species, and further studies should investigate the influence of species-specific λ and extend the AGR method to multi-species. We suspect that the length-scale of variations of the Q-map is determined by the largest λ, at least for systems with a similar number of each atomic species, and different λ may therefore not be beneficial. For this system, energy calculations were performed using DFT with the Perdew-Burke-Ernzerhof (PBE) functional in the GPAW package using a dzp LCAO basis set, constituting a relatively fast full DFT calculation yet much more computationally demanding than the DFTB calculations employed thus far.[57–59]

The timings for different phases of the algorithm on this system are reported in Table I. With the above settings for the DFT calculation, the overall central processing unit (CPU) time is dominated by the DFT calculation, with the building and training phases taking up only 25%–30% of the time, with only a small increase when the Gaussian representation is included.

IX. CONCLUSION

We have shown that augmenting the image representation of the atomistic structure from a one-hot encoding to various forms of Gaussian broadened representations is beneficial for the ASLA, which makes it capable of learning faster from such an input. The conclusion holds for conversion of one input layer to another input layer based on an on-site radial broadening convolution. Providing several input layers with angular information in the form of the power spectrum of the first terms of a circular harmonic expansion of atomic densities, however, does not make the ASLA learn faster. The latter is explained in terms of only some of the extra channels having valuable information, which poses an extra learning challenge, namely, identifying those channels.

We have further shown that employing the Gaussian representation is beneficial for a multi-component system, naphthalene, when studied in a full DFT setting, where the computational

overhead of introducing the Gaussian representation is negligible compared to the cost of the DFT calculations.

ACKNOWLEDGMENTS

We acknowledge support from VILLUM FONDEN (Investigator Grant, Project No. 16562).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[2] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).
[3] W. J. Szlachta and C. G. Bartók, Phys. Rev. B **90**, 104108 (2014).
[4] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, and O. A. von Lilienfeld, J. Chem. Theory Comput. **13**, 5255 (2017).
[5] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, Sci. Adv. **3**, e1603015 (2017).
[6] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg, J. Chem. Phys. **148**, 241733 (2018).
[7] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Phys. Rev. Lett. **104**, 136403 (2010).
[8] A. P. Bartók and G. Csányi, Int. J. Quantum Chem. **115**, 1051 (2015).
[9] J. Behler, J. Chem. Phys. **134**, 074106 (2011).
[10] J. S. Smith, O. Isayev, and A. E. Roitberg, Chem. Sci. **8**, 3192 (2017).
[11] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Nature **559**, 547 (2018).
[12] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, npj Comput. Mater. **5**, 83 (2019).
[13] V. L. Deringer and G. Csányi, Phys. Rev. B **95**, 094203 (2017).
[14] V. L. Deringer, N. Bernstein, A. P. Bartók, M. J. Cliffe, R. N. Kerber, L. E. Marbella, C. P. Grey, S. R. Elliott, and G. Csányi, J. Phys. Chem. Lett. **9**, 2879 (2018).
[15] V. Botu, R. Batra, J. Chapman, and R. Ramprasad, J. Phys. Chem. C **121**, 511 (2017).
[16] R. Jinnouchi, F. Karsai, and G. Kresse, Phys. Rev. B **100**, 014105 (2019).
[17] K. Xia, H. Gao, C. Liu, J. Yuan, J. Sun, H.-T. Wang, and D. Xing, Sci. Bull. **63**, 817 (2018).
[18] S.-D. Huang, C. Shang, P.-L. Kang, and Z.-P. Liu, Chem. Sci. **9**, 8644 (2018).
[19] H. Zhai and A. N. Alexandrova, J. Chem. Theory Comput. **12**, 6213 (2016).
[20] S. Jindal, S. Chiriki, and S. S. Bulusu, J. Chem. Phys. **146**, 204301 (2017).
[21] E. L. Kolsbjerg, A. A. Peterson, and B. Hammer, Phys. Rev. B **97**, 195424 (2018).
[22] M. R. G. Marques, J. Wolff, C. Steigemann, and M. A. L. Marques, Phys. Chem. Chem. Phys. **21**, 6506 (2019).
[23] A. A. Peterson, R. Christensen, and A. Khorshidi, Phys. Chem. Chem. Phys. **19**, 10978 (2017).
[24] A. A. Peterson, J. Chem. Phys. **145**, 074106 (2016).
[25] A. P. Bartók, R. Kondor, and G. Csányi, Phys. Rev. B **87**, 184115 (2013).
[26] F. A. Faber, A. S. Christensen, B. Huang, and O. A. von Lilienfeld, J. Chem. Phys. **148**, 241717 (2018).
[27] G. Ferré, T. Haut, and K. Barros, J. Chem. Phys. **146**, 114107 (2017).
[28] T. Xie and J. C. Grossman, Phys. Rev. Lett. **120**, 145301 (2018).
[29] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, J. Chem. Phys. **148**, 241722 (2018).
[30] E. G. del Río, J. J. Mortensen, and K. W. Jacobsen, Phys. Rev. B **100**, 104103 (2019).
[31] N. Bernstein, G. Csányi, and V. L. Deringer, npj Comput. Mater. **5**, 99 (2019).
[32] L. Zhang, D.-Y. Lin, H. Wang, R. Car, and W. E, Phys. Rev. Mater. **3**, 023804 (2019).
[33] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, and A. R. Oganov, Phys. Rev. B **99**, 064114 (2019).
[34] M. K. Bisbo and B. Hammer, Phys. Rev. Lett. **124**, 086102 (2020).
[35] T. L. Jacobsen, M. S. Jørgensen, and B. Hammer, Phys. Rev. Lett. **120**, 026102 (2018).
[36] X. Chen, M. S. Jørgensen, J. Li, and B. Hammer, J. Chem. Theory Comput. **14**, 3933 (2018).
[37] M. S. Jørgensen, M. N. Groves, and B. Hammer, J. Chem. Theory Comput. **13**, 1486 (2017).
[38] M. S. Jørgensen, U. F. Larsen, K. W. Jacobsen, and B. Hammer, J. Chem. Phys. A **122**, 1504 (2018).
[39] M. Todorović, M. U. Gutmann, J. Corander, and P. Rinke, npj Comput. Mater. **5**, 35 (2019).
[40] C. J. Pickard, Phys. Rev. B **99**, 054102 (2019).
[41] K. H. Sørensen, M. S. Jørgensen, A. Bruix, and B. Hammer, J. Chem. Phys. **148**, 241734 (2018).
[42] S. Chiriki, M.-P. Christiansen, and B. Hammer, Phys. Rev. B **100**, 235436 (2019).
[43] M. S. Jørgensen, H. L. Mortensen, S. A. Meldgaard, E. L. Kolsbjerg, T. L. Jacobsen, K. H. Sørensen, and B. Hammer, J. Chem. Phys. **151**, 054111 (2019).
[44] G. N. C. Simm, R. Pinsler, and J. M. Hernández-Lobato, "Reinforcement learning for molecular design guided by quantum mechanics," arXiv:2002.07717 [stat.ML] (2020).
[45] S. A. Meldgaard, H. L. Mortensen, M. S. Jørgensen, and B. Hammer, J. Condens. Matter. Phys. **32**, 404005 (2020).
[46] I. Wallach, M. Dazamba, and A. Heifets, "AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery," arXiv:1510.02855 (2015).
[47] W. Torng and R. B. Altman, BMC Bioinf. **18**, 302 (2017).
[48] M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, and D. R. Koes, J. Chem. Inf. Model. **57**, 942 (2017).
[49] D. Kuzminykh, D. Polykovskiy, A. Kadurin, A. Zhebrak, I. Baskov, S. Nikolenko, R. Shayakhmetov, and A. Zhavoronkov, Mol. Pharm. **15**, 4378 (2018).
[50] H. L. Mortensen, S. A. Meldgaard, M. K. Bisbo, M.-P. V. Christiansen, and B. Hammer, "Atomistic structure learning algorithm with surrogate energy model relaxation," arXiv:2007.07523 (2020).
[51] B. Aradi, B. Hourahine, and T. Frauenheim, J. Phys. Chem. A **111**, 5678 (2007).
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 27–30 June 2016 (IEEE, 2016), pp. 770–778.
[53] D. E. Worrall, S. J. Garbin, D. Turmurkhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," arXiv:1612.04642 (2017).
[54] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, and K. Kohlhoff, "Tensor-field networks: Rotation and translation-equivariant neural networks for 3D point clouds," arXiv:1802.08219 (2018).
[55] M. Eickenberg, G. Exarchakis, M. Hirn, and S. Mallat, in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), pp. 6540–6549.
[56] M. Abadi, A. Agarwal, P. Barham *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems" (2015), software available from www.tensorflow.org.
[57] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, Phys. Rev. B **71**, 035109 (2005).
[58] A. H. Larsen, M. Vanin, J. J. Mortensen, K. S. Thygesen, and K. W. Jacobsen, Phys. Rev. B **80**, 195112 (2009).
[59] J. Enkovaara, C. Rostgaard, J. J. Mortensen *et al.*, J. Phys.: Condens. Matter **22**, 253202 (2010).