npj | Computational Materials

www.nature.com/npjcompumats

ARTICLE OPEN

Check for updates

Bayesian force fields from active learning for simulation of inter-dimensional transformation of stanene

Yu Xie[1✉], Jonathan Vandermause[1], Lixin Sun[1], Andrea Cepellotti[1] and Boris Kozinsky[1,2✉]

We present a way to dramatically accelerate Gaussian process models for interatomic force fields based on many-body kernels by mapping both forces and uncertainties onto functions of low-dimensional features. This allows for automated active learning of models combining near-quantum accuracy, built-in uncertainty, and constant cost of evaluation that is comparable to classical analytical models, capable of simulating millions of atoms. Using this approach, we perform large-scale molecular dynamics simulations of the stability of the stanene monolayer. We discover an unusual phase transformation mechanism of 2D stanene, where ripples lead to nucleation of bilayer defects, densification into a disordered multilayer structure, followed by formation of bulk liquid at high temperature or nucleation and growth of the 3D bcc crystal at low temperature. The presented method opens possibilities for rapid development of fast accurate uncertainty-aware models for simulating long-time large-scale dynamics of complex materials.

npj Computational Materials (2021)7:40 ; https://doi.org/10.1038/s41524-021-00510-y

INTRODUCTION

Density functional theory (DFT) is one of the most successful methods for simulating condensed matter thanks to a reasonable accuracy for a wide range of systems. Ab initio molecular dynamics (AIMD) offers a way to simulate the atomic motion using forces computed at the DFT level. Unfortunately, computational requirements limit the timescale and size of AIMD simulations to a few hundred atoms for a few hundred picoseconds of time, precluding investigation of phase transitions and heterogeneous reactions. Such large-scale molecular dynamics (MD) simulation must resort to empirically derived analytical interatomic force fields with fixed functional form[1–4], trading accuracy and transferability for larger length and timescales. Classical analytical force fields often do not match the accuracy of ab initio results, limiting simulations to describing results qualitatively at best or, at worst, deviating from the correct behavior. In order to broaden the reach of computational simulations, it would be desirable to compute forces with ab initio accuracy at the same cost of classical interatomic force fields.

In recent years, machine learning (ML) algorithms emerged as powerful tools in regression and classification problems. This interest has inspired several works to develop ML algorithms for interatomic force fields, for example neural networks (NNs)[5–12], MTP[13–15], FLARE[16,17], GAP[18–21], SNAP[22], SchNet[23–25], DeePMD[26–28], among others. All these ML potentials can make predictions at near ab initio accuracy, while greatly reducing the computational cost compared to DFT.

However, most ML models have no predictive distributions, which provide uncertainty for energy/force predictions. Without uncertainty, the training data are generally selected from ab initio calculations via a manual or random scheme. Determining the reliability of the force fields then becomes difficult, which could result in untrustworthy configurations in the MD simulations. Therefore, uncertainty quantification is a highly desirable capability[29,30]. NN potentials, e.g., ANI[11,31] uses ensemble disagreement as an uncertainty measure. This statistical approach is, however, not guaranteed to yield reliable calibrated uncertainty. In addition, NN approaches usually require tens of thousands of data for training, and are a few orders of magnitude slower than analytical force fields. Bayesian models are promising for uncertainty quantification in atomistic simulations since they have an internal principled uncertainty quantification mechanism, the variance of the posterior prediction distribution, which can be used to keep track of the error on forces during a MD run. For instance, Jinnouchi et al.[32,33] used high-dimensional SOAP descriptors with Bayesian linear regression[19]. Gaussian process (GP) regression[18,19,34–36] is a Bayesian method that has been shown to learn accurate forces with relatively small training data sets. Bartok et al.[21] used GP uncertainty with the GAP/SOAP framework to obtain only qualitative estimates of the force field's accuracy. Recently Vandermause et al.[16] demonstrated a GP-based Bayesian active learning (BAL) scheme in the FLARE framework, utilizing rigorously principled and quantitatively calibrated uncertainty, applying it to a variety of multi-component materials.

In the most common form, however, GP models require using the whole training data set for prediction of both the force and uncertainty, meaning that the computational cost of prediction grows linearly with the size of the training set, and so accuracy increases together with computational cost. In complex multi-component structures, more than $O(10^2)$ training structures or $O(10^3)$ local environments are typically required to construct an accurate GP model, which makes predictions slow[37,38]. Because of the linear scaling of GP, the on-the-fly training becomes slower as more data are collected. This precludes the active learning of GP on larger system sizes, which may be needed due to finite size effects, as well as longer timescales needed to explore phase space more thoroughly. To accelerate the on-the-fly training, fast and lossless mappings as approximations of both the force and uncertainty are desired, such that the mappings can replace GP to make predictions during the BAL.

The force mapping has been addressed by Glielmo et al.[34,34–36] and Vandermause et al.[16]. They noted that, for a suitable choice of

[1]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. [2]Robert Bosch LLC, Research and Technology Center, Cambridge, MA, USA. ✉email: xiey@g.harvard.edu; bkoz@seas.harvard.edu

the n-body kernel function, it is possible to decompose GP force/energy prediction into a summation of low-dimensional functions of atomic positions. As a result, one can construct a parametric mapping of the n-body kernel function combined with a fixed training set. This approach reaches a constant scaling, which increases the speed of the GP model without accuracy loss. We also present the formalism of the force mapping in "Methods".

Since the uncertainty plays a central role in the on-the-fly training[16] and also suffers from the linear-scaling complexity with respect to the training set size, it is desirable to have a similar mapping of the uncertainty as that of the force. However, this was not attempted in refs. [16,34–36], since the mathematical form of the predictive variance results in decomposition with twice the dimension of the corresponding mean function, dramatically increasing the computational cost of their evaluation with spline interpolation. Thus, to date there is no method that combines high accuracy, modest training data requirement and fast prediction of both forces and their principled Bayesian uncertainty. In this work, we introduce a dimensionality reduction technique for the uncertainty mapping, such that the interpolation can be done on the same dimensionality as the force mapping, enabling development and application of efficient Bayesian force field (BFF) models for complex materials. The mathematical formalism of the uncertainty mapping is presented in "Methods".

In this article, we present an accurate mapping of both force (mean) and uncertainty (variance) implemented as the mapped Gaussian process (MGP) method. As a result, the MGP method benefits from the capability of quantifying uncertainty, while at the same time retaining cost independent of training size. The original BAL with GP can then be accelerated by the fast evaluation of forces and uncertainties of MGP. The training can also be extended to larger system sizes and longer timescales that are challenging for the full GP. To illustrate the performance for large-scale dynamics simulations, we incorporate the MGP force field with mean-only mapping in the parallel MD simulation code LAMMPS[39], and apply it to the investigation of phase transformation dynamics. The MGP force field is shown to be efficient for large-scale (million atom) simulations, achieving speeds comparable to classical analytical force fields, several orders of magnitude faster than available NN or full GP approaches.

As a test application, we focus on stanene, a 2D material that has recently gathered attention as a quantum spin Hall insulator[40]. Moreover, the only published force field for stanene is, to the best of our knowledge, the bond-order potential by Cherukara et al.[41], which is fitted to capture stanene's low-temperature crystalline characteristics but, as common to many empirical interatomic potentials, suffers in accuracy near the melting temperature. In particular, we show that MGP is capable of rapidly learning to describe a wide range of temperatures, below and around the melting temperature, and that we can efficiently monitor the uncertainty on forces at each time step of the MD, and use this capability to iteratively increase the accuracy of the force field by hierarchical active training. Using parallel simulations of large-scale structures, we characterize the unusual phase transition, where the 2D monolayer transforms to bcc bulk Sn at the temperature of 200 K, and melts to ultimately form a 3D liquid phase at the temperature of 500 K.

RESULTS

Accelerated Bayesian active learning with MGP

In a MD simulation, it is likely that the system will evolve to atomic configurations unseen before, and are far from those in the training set. In this situation, the uncertainties of the predictions will grow to large values, which may be considered unsatisfactory. Therefore, it is desirable to obtain an accurate estimate of the forces for such configurations using a relatively expensive first principles calculation, and add this new information to the training set of the GP regression model.

This procedure is referred to as BAL, where training examples are added on the fly as more information is obtained about the problem. Bayesian ML models such as GP are particularly well-suited to such uncertainty-based active learning approaches, as they provide a well-defined probability distribution over model outputs, which can be used to extract rigorous estimates of model uncertainty for any test point[42,43].

Here we adopt BAL to achieve automatic training of models for atomic forces, expanding on our earlier workflow[16]. This way the accuracy of the GP model increases with time, particularly in the configuration regions that are less explored and likely to be inaccurate.

As mentioned above, GP regression cost scales linearly with the training set size, so the prediction of forces and uncertainties becomes more expensive as the BAL algorithm keeps adding data to the training set, hindering simulations of complex materials systems. The MGP approach is an essentially exact mapping of both mean and variance of the predicted forces from a full GP model for a fixed training set to a low-dimensional spline model. This approach has the ability to accelerate prediction of not only the mean, but also GP's internal uncertainty without loss of accuracy. MGP is incorporated into the BAL workflow as depicted in Fig. 1. In this scheme, the mapping is constructed every time the GP model is updated. Atomic forces and their uncertainties for MD are produced by MGP, reducing the computational cost of the full original GP prediction by orders of magnitude. We stress that not only is the MGP BFF faster than the original GP, but also its speedup is more pronounced as the training data set size N_T increases. This is especially critical in complex systems with multiple atomic and molecular species and phases, where more training configurations are needed.

In addition, fast evaluation of forces and their variances enables training the force field model on larger system sizes. Therefore, we can train the BFF using a hierarchical scheme, i.e., for different system sizes, we can run several iterations of active learning. First, a small system size is used to train the BFF using the BAL workflow. Then, we perform BAL on a larger system size, where DFT calculations are more expensive but are needed less frequently, in order to capture potentially new unexplored configurations such as defects that are automatically identified as high uncertainty by the model of the previous step. We note that DFT calculations scale as the 3rd power of the number of electrons in the system, so it is efficient to perform most of the training at smaller sizes, and to only require re-training on fewer large-scale structures. This way the hierarchical training scheme helps to overcome finite size effects and explore phenomena that cannot be captured with small system sizes, such as phase transformations.

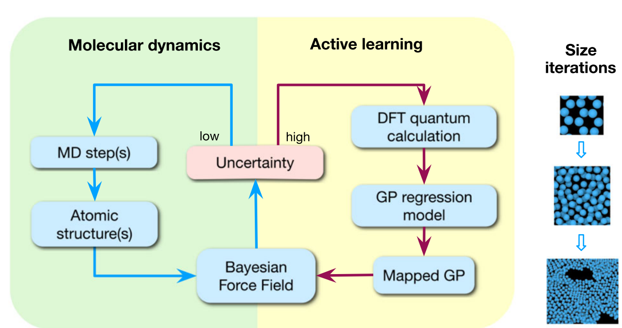

Fig. 1 Bayesian active learning (BAL) workflow with MGP. Iterations of BAL can be run for different system sizes to refine the force field.

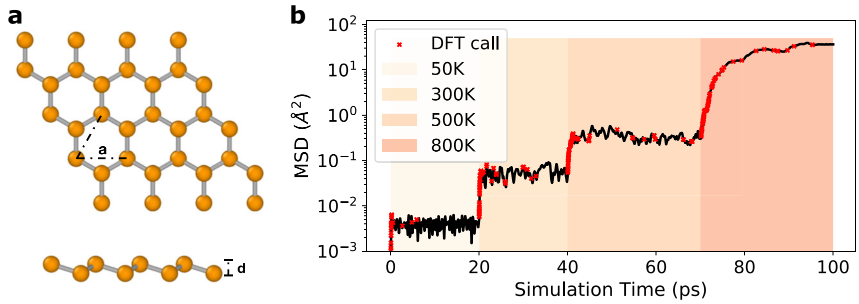

Fig. 2 Bayesian active learning of stanene. **a** Pristine stanene, top view and front view. **b** Atomic mean square displacement (MSD) of the on-the-fly training trajectory. The solid black line shows the MSD of the 32 atoms, the color blocks indicate time intervals corresponding to different temperatures, and the red dots represent events when DFT calculations are called as the uncertainty predictions exceed the data acquisition decision threshold and new data are added to the training set.

Case study: 2D to 3D transformation of stanene

To illustrate our method, we apply the mapped on-the-fly workflow (Fig. 1) to study the phase transformation of stanene, a two-dimensional slightly buckled honeycomb lattice of tin. This structure has received much attention recently due to its unique topological electronic properties. Stanene can be synthesized by molecular beam epitaxy[44] and has been shown to have topologically protected states in the band gap[45]. Moreover, tri-layer stanene is a superconductor[46]. As a demonstration of our methodology, here we want to focus on the less-studied aspects of thermodynamic stability and phase transition mechanisms of free-standing stanene. We note that a substrate is currently always used in the growth of stanene monolayers, and the interaction between the monolayer and the substrate will be investigated in a future work. By studying the intrinsic thermodynamics properties of the free-standing stanene, we are able to examine its stability and decomposition timescales as a function of temperature, which are relevant for device engineering.

Despite the intense interest, capability is still lacking for accurate modeling of thermodynamic properties and phase stability of stanene, and most other 2D materials, for which ~nm scale supercells are required. Such simulations are beyond the ability of DFT, and to the best of our knowledge, the only published classical force field for stanene is a bond-order potential (BOP)[41]. Our focus is on developing an accurate potential to model the 2D melting process, which is a notoriously difficult problem to tackle with existing classical force fields. In fact, describing the melting mechanism of a 2D material is a highly nontrivial problem, since it can be characterized by specific behaviors such as defect formation and nucleation. For example, the melting of graphene is thought to occur in a temperature range between 4000 and 6000 K[47–49], and seems to be characterized by the formation of defects, while above 6000 K, the structure melts directly into a totally disordered configuration. Since bulk tin melts at a much lower temperature (~500 K) than graphite (~4300 K), we expect that stanene will transform at a relatively low temperature. However, little is known about its melting mechanism and transition temperature. The MGP method is uniquely suited to describe the transformation processes, as it is scalable to large systems and capable of accurately describing any configuration that is sufficiently close to the training data.

To train the force field for stanene, we start by training the MGP BFF using the BAL loop on a small system size, and gradually increase the size of the simulation, iteratively improving upon the force field developed for the smaller size.

The DFT settings and the hyperparameters of GP are shown in Supplementary Table 1. The parameters of MGP model are shown in Supplementary Table 2. The atomic configurations are plotted by OVITO[50].

The first iteration of training is started with a $4 \times 4 \times 1$ supercell of stanene (32 atoms in total) and a 20-ps long MD simulation at a temperature of 50 K. Subsequently, we increase the temperature of the system at 300, 500, and 800 K by velocity rescaling, and let the system evolve for 20, 30, and 30 ps, respectively. This way we efficiently augment the training set with relevant configurations. During the simulation, DFT was called whenever the uncertainty on any single force component exceeded the current optimized noise parameter of the GP, after which the $N = 1$ atomic environment with the highest force uncertainty was added to the training set. Figure 2b shows the mean square displacement of atoms during the on-the-fly BAL training. As one would expect, the DFT is called multiple times at the beginning of the MD simulation, since the model needs to build an initial training data set to make force predictions. In addition, the DFT is called whenever temperature is increased, as the trajectory explores new configurations that have not been visited before. Clearly, the variance increases rapidly when temperature is increased.

In this small simulation, the stanene monolayer decomposes at 800 K, where Sn atoms start moving around the simulation cell. This allows us to augment the training set with such disordered configurations. However, the small size of the simulation cell does not yet allow us to draw conclusions about the melting process of stanene: for this, we need a larger simulation cell.

We continue with our hierarchical active learning scheme to refine the force field, but retaining the training data from the previous smaller MD run. We now increase the number of atoms in the simulation, which allows us to explore additional atomic configurations and improve the quality of the model. Since the system becomes larger and more computationally expensive, instead of updating the model at each learning step, we update it at the end of each MD simulation. This "batching" approach is one of many possible training schemes and is chosen for reasons of efficiency. The procedure is as below:

(1) Run MD simulation on a system of a given size with MGP;
(2) Predict mapped uncertainties for the frames from the MD trajectory;
(3) Select frames of highest uncertainties in different structures (crystal, transition state, and disordered phase) for additional DFT calculations;
(4) Add atomic environments from those frames based on uncertainty to the training set of GP;
(5) Construct a new MGP BFF using the updated GP;
(6) Repeat the process for same or larger simulation using the new BFF.

The above steps form one iteration, and can be run several times to refine the force field. In our force field training for stanene, we performed one iteration of 32 atoms as described before, and two iterations all on the system size of 200 atoms at 500 K, a relatively high temperature to explore diverse structures in the configuration phase space. We denote the MGP BFF from the on-the-fly training of 32 atoms as MGP-1, the ones from the

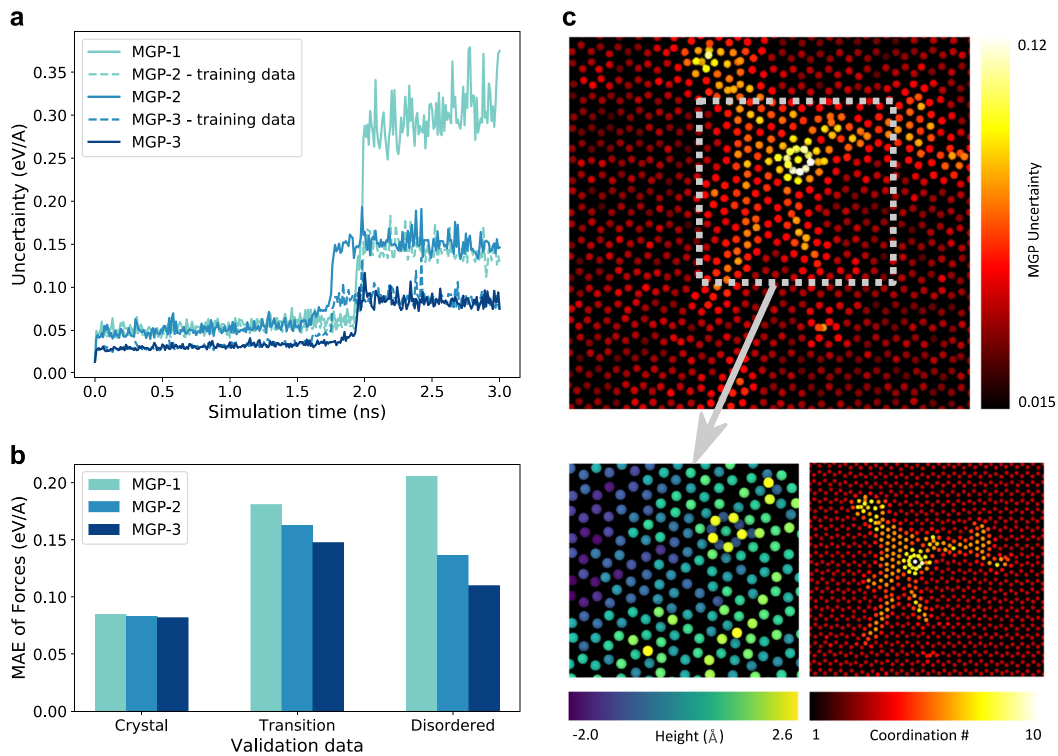

Fig. 3 Uncertainty and accuracy in the Bayesian active learning. a Uncertainty of the force predictions in the whole MD trajectory decreases in three iterations. **b** Mean absolute errors (MAE) of MGP forces against DFT in three iterations, validated on atomic structures of different phases. **c** Upper: a snapshot of 2048 atoms from MD simulation at 500 K, colored by MGP uncertainty. The outlined area has highest predicted uncertainty, since it contains a defect and a compact region not known to the first MGP model. Lower-left: zoomed-in structure of the region of high uncertainty, colored by atomic heights (*z*-coordinate) to show the bilayer structure of the defect. Lower-right: the same snapshot colored by coordination number, which shows strong correlation with the uncertainty measure.

two iterations of training on 200 atoms as MGP-2 and MGP-3. MD are simulated by MGP-1, 2, 3, and the predicted uncertainties of the frames from the three MD trajectories predicted by MGP BFF are shown in Fig. 3a. First, there is a drastic increase of uncertainty within each trajectory at a certain time. This increase corresponds to the transition from crystal lattice to a disordered phase, so the uncertainty acts as an automatic indicator of the structural change. Next, the uncertainties of MGP-1, 2, 3 decrease iteration by iteration, indicating our model becomes more confident as it collects more data. The mean absolute error of force predictions of MGP BFF against DFT is investigated, as presented in Fig. 3b, showing that MGP BFF becomes also more accurate with each subsequent training iterations. The accuracy of force prediction from MGP force field is also much higher than BOP[41] for different phases of stanene, as shown in Supplementary Table 4.

It is worth noting that in the transition process of the simulation by MGP-1, there appear a bilayer stacked defect and a specific compact triangular lattice around the defect. It is a visibly new ordered structure different from the hexagonal stanene, and is thus missing in the initial training set, as shown in Fig. 3c. The uncertainty is low in honeycomb lattice, as expected, and becomes high around the bilayer defect, exhibiting also a strong correlation with the coordination number. Thus, the mapped uncertainty is an accurate automatic indicator of the novelty of the new configurations. We note that after iterative model refinement by the hierarchical training scheme, the triangular lattice no longer appears in the simulation. This example illustrates how the hierarchical active learning scheme helps avoid qualitatively incorrect results that arise due to the a priori unknown relevant configurations. In contrast, a single-pass training of a force field based on manually chosen configurations is very likely to result in nonphysical predictions.

To investigate the full phase transition mehanism of stanene, we consider low temperature (below 200 K) and high temperature (above 500 K) values, corresponding to the crystal and liquid phase for bulk tin. In studying the graphite–diamond transition, Khaliullin et al.[51] identified that nucleation and growth of the disordered phase play key roles in the phase transition process. We also find that in stanene, the phase transition begins with the appearance of defects and the consequent nucleation and growth of the disordered phase that eventually transforms the structure.

From DFT ab initio calculations, the ground state energy of honeycomb monolayer is ~0.38 eV/atom higher than the bulk, so the 2D monolayer structure is expected to be meta-stable. The full transformation described below is therefore a sequence of 2D melting, followed by a kinetically controlled transition to the bulk phase of lower free energy. We set up the simulation of stanene monolayer of size 15.6 × 18.1 nm (3000 atoms), and run MD in the NPT ensemble for 3 ns at the temperature of 200 K. We discover that the ordered honeycomb monolayer loses long-range order and undergoes densification, and finally rearranges into the bcc bulk structure spontaneously, following three transformation steps. At first, the atoms vibrate around the honeycomb lattice sites, and the monolayer remains in its ordered 2D phase (Fig. 4a—0.3 ns). The main deviation from the ideal lattice is the out-of-plane rippling due to membrane fluctuations. The fluctuating ripples are similar to what was reported in simulations of graphene[47–49]. Next, double-layer stacking defects appear due to buckling and densification. These defect regions grow and quickly expand to the whole system, indicative of a first order phase transition in 2D. The atomic configuration appears to be a "disordered" phase (Fig. 4a—0.8 ns). In this stage, the structure becomes denser and increasingly three-dimensional driven by energetic gains from increasing Sn atom coordination, with the

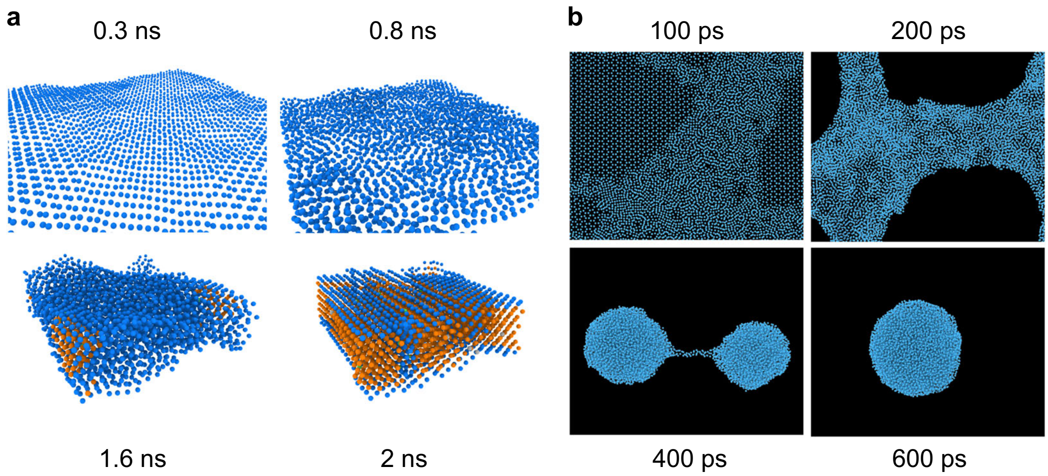

Fig. 4 Large-scale molecular dynamics simulation of the inter-dimensional transformation of stanene. a The monolayer–bulk transition at ~200 K. Using common neighbor analysis, the orange atoms are identified as bcc sites, and atoms of other colors are identified as other sites. **b** Frames from ~500 K NPT simulation with 10,000 atoms (39.4 × 22.7 nm) at 100, 200, 400, and 600 ps, showing the melting process of the monolayer.

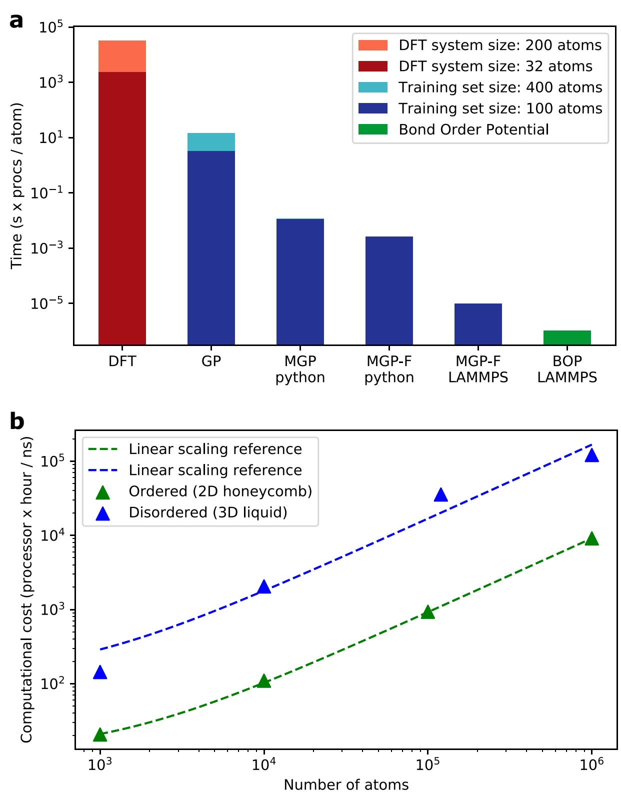

Fig. 5 Computational cost and scaling. a The comparison of prediction time (s × processors/atom) between DFT, full GP, MGP (Python, with force and uncertainty), MGP-F (Python, with force only), MGP (LAMMPS, with force only), and BOP (LAMMPS). **b** The scaling of MGP in LAMMPS with respect to different system sizes, in different phases. Green: 2D honeycomb lattice. Blue: 3D dense liquid.

unit cell dimensions shrinking. Finally, patches of bcc lattice nucleate during the atomic rearrangement (Fig. 4a—1.6 ns), and those patches grow until the whole system forms the crystal bcc slab and atoms start vibrating around the lattice sites (Fig. 4a—2 ns).

The stable phase of bulk tin below 286.35 K is expected to be the α-phase with the diamond structure, which is different from what we observe here. The reason for the formation of the bcc phase is kinetic rather than thermodynamic, caused by a lower free energy barrier separating the 3D disordered structure from the bcc phase than the α-phase, making the former phase more kinetically accessible. One well-known example is the graphite–diamond transition, in which the meta-stable hexagonal diamond (HD) is obtained from hexagonal graphite (HG) in most laboratory synthesis, instead of the more thermodynamically stable cubic diamond (CD)[52–56]. Computational study[51] reveals that the higher in-layer distortions result in higher energy barrier for HG–CD than HG–HD transformation, under most of the experimental conditions. We observe that the 2D stacking order-disorder and the 3D crystallization transformation steps proceed via nucleation and growth, where the defect nuclei grow and turn the whole system into the intermediate disordered dense phase, while in the second step the bcc nuclei appear from the disordered configuration and grow, turning the system into the bcc crystal phase. Our simulation result implies the meta-stability of the free-standing stanene monolayer. The evidences provided by the MD simulation are consistent with the experimental fact that all the existing growths of the stanene monolayer are achieved on substrates[3,44–46,57–59]. The role that the substrate plays in stabilizing the two-dimensional stanene monolayer is thus of great interests and realistic significance, and will be investigated in our future work.

To investigate the melting behavior at higher temperature, we focus on a 600 ps long MD run that starts from a pristine sample of stanene with size 39.4 × 22.7 nm (10,000 atoms) at ~500 K. A few frames of this simulations are shown in Fig. 4b. The melting process can also be identified as three stages. The first stage is the development of large out-of-plane ripples in the free-standing 2D lattice. In the second stage, at around 100 ps, double-layer defects in the monolayer form and agglomerate, causing densification. After 200 ps, densification due to large patches of disordered structures lead to the formation of holes, and a large void area grows. We observe breaking of the two-dimensionality of the crystal and formation of stacked configurations where atoms climb over other atoms of 2D stanene. In the third stage, the disordered stacked configurations shrink into spherical liquid droplets connected by a neck (400 ps). Subsequently the droplets merge to form a large sphere, minimizing the surface energy

Table 1. Performance in LAMMPS.

System size (atoms)	1,000,000
CPU number	256
Thermostat	NVT
Temperature (K)	100
BOP cutoff (Å)	6.0
MGP cutoff (Å)	7.2
BOP speed (atom·timestep/processor/s)	$\sim 2.3 \times 10^6$
MGP speed (atom·timestep/processor/s)	$\sim 5.0 \times 10^5$

(600 ps). In Supplementary Fig. 3, we include more discussions and animations of the transition processes.

Performance

In this section, we discuss the computational cost of the MGP model in application to large-scale MD simulations, and its advantages and limitations. Timing results are shown in Fig. 5a, comparing different models including DFT, full GP (without mapping; implemented in Python), MGP (with mapped force and uncertainty; Python), MGP-F (MGP with force prediction only; Python), MGP (force only; implemented in LAMMPS), and BOP (LAMMPS force field). For DFT, we tested prediction time for system sizes of 32 and 200 atoms, where computational cost grows as the cube of the system size. The other algorithms' cost scales linearly with system size, since they rely on local environments with finite cutoff radius, we measure the same per atom computational cost for both system sizes.

It is important to highlight the acceleration of the MGP relative to the original full GP. We present the timings of GP and MGP built from different training set sizes (100 and 400 training data) in Fig. 5, where the GP scales linearly. MGP-F (without uncertainty) and MGP (with uncertainty, fixed rank) are independent of the training set size. In the stanene system, MGP BFF is two orders of magnitude faster than GP with $O(10^2)$ training data points when both force and variance are predicted, and the speedup becomes more significant as the GP collects more training data, as reported in Supplementary Fig. 4. We also note that although the prediction of MGP gets rid of the linear scaling with training set size, the construction of the mappings does scale linearly with training set size, since it requires GP to make prediction on each grid point.

The cost of the MGP force field implemented in LAMMPS (force only) is comparable to that of empirical interatomic BOP force field, since both BOP and MGP models consider interatomic distances within the local environment of a central atom and thus have similar computational cost at the same cutoff radius. For accurate simulations, however, the MGP uses a larger cutoff radius (7.2 Å) than what is used by the BOP (6 Å), and thus a simulation with our MGP force field for stanene is slower by a factor of 4–7 approximately, see our test of 1 million atoms in Table 1. The computational cost of the MGP force field in LAMMPS as a function of the system size is shown in Fig. 5b. The prefactor of the linear dependence depends on the number of pairs and triplets within a local environment. The scaling factor for the 2D honeycomb lattice structure is lower than for the dense 3D liquid, because each atom in the 3D liquid has more neighbors than the honeycomb lattice within the same cutoff radius.

The hierarchical active learning scheme adopted here is an efficient path to systematically improve the force field. However, it must be noted that the training is still limited by the system sizes that can be tackled with a DFT calculation. In fact, DFT calculations with thousands of atoms are unaffordable. To circumvent this problem, regions of high uncertainties may be cropped out of the large simulation cell, in order to perform DFT calculation on a smaller subset of atoms with a relevant structure. In this direction, we note that uncertainty is a local predicted property in MGP BFF, indicating how each atom contributes to the total uncertainty. In Fig. 3c, we show that the largest uncertainty in an MD frame of 2048 atoms comes from regions close to defects, since these configurations are missing in the training set for that particular model. Therefore, one may use the uncertainty to automatically select a section of the crystal close to the high-uncertainty region, and use a DFT simulation to add a training data point to the MGP. Automatically constructing such structures remains an open challenge in complex materials. However, access to the local uncertainty for each force prediction already allows for assessment of reliability of simulations, and is central to the paradigm of BFFs.

Finally, we mention that the descriptive power of our current BFF is limited due to the lack of full many-body interactions in GP kernel, since the n-body kernel only uses low-dimensional functions of crystal coordinates. In this work, we have implemented the MGP for 2- and 3-body interactions, which, for the mapping problem, translate to interpolation problems in three dimensions. The approach may be extended to higher interaction orders at the expense of computational efficiency, but it becomes expensive in terms of both computation and storage requirements for splines on uniform grids to reach a satisfactory accuracy. Many-body descriptors can be taken into account to increase the descriptive power. To map it in the same way as the methodology we introduced, it is necessary to have a descriptor such that it can be decomposed into low-dimensional functions. For example, the SOAP approach of refs. [32,33] cannot be mapped using the methodology followed in this work, since it relies on a high-dimensional descriptor. In particular, the interpolation procedure may be too computationally expensive to be a viable solution.

DISCUSSION

In conclusion, we present an extended method for mapping the GP regression model with many-body kernels, where we accurately parameterize both the force and its variance obtained from the original Bayesian model as functions of atomic configurations. The mapped forces and their variance are then utilized in an active learning workflow to obtain a machine-learned force field that includes its own uncertainty, the first realization of a fast BFF. The resulting model has near-quantum accuracy and computational cost comparable to empirical analytical interatomic potentials. Large-scale simulations are used as a demonstration to investigate the microscopic details of inter-dimensional transformation behavior of 2D stanene into 3D tin. We discover that the transformation proceeds by nucleation of bilayer defects, densification due to continued disordered multi-layer stacking, and finally conversion into either crystalline or molten bulk tin, depending on temperature. This application shows that we can actively monitor the uncertainties of force predictions during MD simulations and iteratively improve the model as the material undergoes reorganization, exploring diverse structures in the configuration phase space. The ability to reach simulation sizes of over 1 million atoms is promising for application of ML force fields to study phase diagrams and transformation dynamics of complex systems, while automated active learning of force fields opens a way to accelerate wide-range material design and discovery.

METHODS

Background: Gaussian process force field

In this section, we summarize the GP model introduced in ref. [16]. From here on, we use bold letters to denote matrices/vectors and regular letters for scalars/numbers.

An atomic configuration is defined by atomic coordinates $\boldsymbol{R}_\alpha$, a vector of dimension equal to the number of atoms N_{atoms}, where α labels the three

Cartesian directions $\alpha = x, y, z$. While in general the crystal may contain atoms of different species, we here consider only a single species case for the sake of simplicity. The results can be extended to the multi-component case, which is discussed in Supplementary Method 1.

The objective of the GP regression model is to predict the forces $\boldsymbol{F}_\alpha$ for a given atomic configuration $\boldsymbol{R}_\alpha$. To this aim, we associate the forces of an atom i, $F_{i,\alpha}$, to the local environment around this atom ρ_i. $F_{i,\alpha}$ can then be computed by comparing its environment with a training data set of atomic environments with known forces, which is quantified by distance d between the target environment and the training set.

The atomic environment ρ_i of the atom i is defined as all the atoms that are within a sphere of a cutoff radius r_{cut} centered at the atom i. We proceed by partitioning this atomic environment ρ_i into a set of n-atom subsets, denoted by $\rho_i^{(n)}$. A distance $d^{(n)}$ can be defined to quantify the difference between environments. We show examples of $n = 2$ and $n = 3$ on the construction of $\rho_i^{(n)}$ and $d^{(n)}$. For $n = 2$, $\rho_i^{(2)}$ is a set of pairs p, between central atom i and another atom i' in ρ_i,

$$\rho_i^{(2)} = \{p = (i, i'); i' \in \rho_i, i' \neq i\}; \tag{1}$$

To each atomic pair p we associate an interatomic distance r_p between the two atoms. The distance metric between two pairs p and q is then defined as $(r_p - r_q)^2$. And the distance metric between two atomic environments $\rho_i^{(2)}$ and $\rho_j^{(2)}$ is defined as:

$$\left(d_{ij}^{(2)}\right)^2 = \sum_{p \in \rho_i^{(2)}, q \in \rho_j^{(2)}} (r_p - r_q)^2. \tag{2}$$

This distance $d^{(2)}$ will be used to determine the two-body contribution to the atomic forces, for the purposes of constructing the kernel function.

This construction can be extended to higher-order interatomic interactions. In this work, we also include three-body interactions, corresponding to $n = 3$. Similar to the two-body case, the set $\rho_i^{(3)}$ of triplets p between the central atom i and two other atoms i' and i'' in ρ_i is defined as:

$$\rho_i^{(3)} = \{p = (i, i', i''); i', i'' \in \rho_i, i' \neq i, i'' \neq i, i' \neq i''\}. \tag{3}$$

This time, each triplet p is characterized by a vector $\boldsymbol{r}_p = \{r_{p,1}, r_{p,2}, r_{p,3}\}$, which represents the three interatomic distances of this atomic triplet p. Including permutation, the distance metric between two triplets p and q can be defined as $\sum_{u,v}(r_{p,u} - r_{q,v})^2$. And the 3-body distance metric between two atomic environments $\rho_i^{(3)}$ and $\rho_j^{(3)}$ is

$$\left(d_{ij}^{(3)}\right)^2 = \sum_{p \in \rho_i^{(3)}, q \in \rho_j^{(3)}} \sum_{u,v} (r_{p,u} - r_{q,v})^2 . \tag{4}$$

Having now defined a distance between atomic environments, we can build a GP regression model for atomic forces. We define the equivariant force–force kernel function to compare two atomic environments as:

$$k_{\alpha\beta}(\rho_i, \rho_j) = \sum_n k_{\alpha\beta}^{(n)}(\rho_i^{(n)}, \rho_j^{(n)}) = \sum_n \frac{\partial^2 k^{(n)}(\rho_i^{(n)}, \rho_j^{(n)})}{\partial R_{i\alpha} \partial R_{i\beta}} , \tag{5}$$

where α and β label Cartesian coordinates, and $\frac{\partial}{\partial R_{i\alpha}}$ indicates the partial derivative of the kernel with respect to the position of atom i. The n-body energy kernel function $k^{(n)}(\rho_i^{(n)}, \rho_j^{(n)})$ is first expressed in terms of the partitions of n atoms of the two atomic environments ρ_i and ρ_j, and is defined as:

$$k^{(n)}(\rho_i^{(n)}, \rho_j^{(n)}) = \sum_{\substack{p \in \rho_i^{(n)} \\ q \in \rho_j^{(n)}}} \tilde{k}^{(n)}(p, q) = \sum_{\substack{p \in \rho_i^{(n)} \\ q \in \rho_j^{(n)}}} \sigma_{\mathrm{sig}}^{(n)} \exp\left[-\frac{\left(d_{ij}^{(n)}\right)^2}{2\left(l^{(n)}\right)^2}\right] \varphi_{\mathrm{cut}}^{(n)} , \tag{6}$$

where $\tilde{k}^{(n)}(p, q)$ is a Gaussian kernel function, built using the definition of distance between n-atoms subsets as described before. The $\sigma_{\mathrm{sig}}^{(n)}$ and $l^{(n)}$ are hyperparameters independent of p and q, that set the signal variance related to the maximum uncertainty and the length scale, respectively. The smooth cutoff function $\varphi_{\mathrm{cut}}^{(n)}$ dampens the kernel to zero as interatomic distances approach r_{cut}. In this work, we chose a quadratic form for the cutoff function

$$\varphi_{\mathrm{cut}}^{(2)}(p, q) = (r_q - r_{\mathrm{cut}})^2 (r_p - r_{\mathrm{cut}})^2, \tag{7}$$

$$\varphi_{\mathrm{cut}}^{(3)}(p, q) = \prod_s (r_p^s - r_{\mathrm{cut}})^2 \prod_t (r_q^t - r_{\mathrm{cut}})^2. \tag{8}$$

The GP regression model uses training data as a reference set of N_T atomic environments. In particular, predictions of force values and variances require a $3N_T \times 3N_T$ covariance matrix

$$\boldsymbol{\Sigma} = \Sigma_{i\alpha, j\beta} = k_{\alpha\beta}(\rho_i, \rho_j) + \sigma_n^2 \delta_{ij} \delta_{\alpha\beta} \tag{9}$$

where σ_n is a noise parameter (indices $i\alpha$ and $j\beta$ are grouped together). We define an auxiliary $3N_T$ dimensional vector $\boldsymbol{\eta}$ with components

$$\eta_{i\alpha} := (\Sigma^{-1})_{i\alpha, j\beta} F_{j\beta} . \tag{10}$$

In order to make a prediction of forces and their errors on a target structure ρ_i, we compute the Gaussian kernel vector $\overline{\boldsymbol{k}}_{i\alpha}$ for atom i and a direction α ($\alpha = x, y, z$), where the components of the vector are kernel distances $(\overline{k}_{i\alpha})_{j\beta} := k_{\alpha\beta}(\rho_i, \rho_j)$ between the prediction $i\alpha$ and the training set points $j\beta$, $j = 1, ..., N_T$, $\beta = x, y, z$. Finally, the predictions of the mean value of the Cartesian component α of the force $F_{i\alpha}$ acting on the atom at the center of the atomic environment ρ_i and its variance $V_{i\alpha}$ are given by ref. [60]:

$$F_{i\alpha} = \overline{\boldsymbol{k}}_{i\alpha}^\top \boldsymbol{\eta} , \tag{11}$$

$$V_{i\alpha} = k_{\alpha\alpha}(\rho_i, \rho_i) - \overline{\boldsymbol{k}}_{i\alpha}^\top \boldsymbol{\Sigma}^{-1} \overline{\boldsymbol{k}}_{i\alpha}. \tag{12}$$

We notice that this formulation has two major computational bottlenecks, that we intend to address in this work. First, the vector $\overline{\boldsymbol{k}}_{i\alpha}$ needs to be constructed for every force prediction; for each atom, the evaluation of $\overline{\boldsymbol{k}}_{i\alpha}$ requires the evaluation of the kernel between the atomic environment and the entire training data set, i.e., N_T calls to the Gaussian kernel. We discuss an approach to speed up the evaluation of $\overline{\boldsymbol{k}}_{i\alpha}$. Second, every variance calculation requires a matrix-vector multiplication that is computationally expensive, and thus will be further discussed in "Methods".

Mapped force field

In GP regression, a kernel function is used to build weights that depend on the distance from training data: training data that are similar to the test data contribute with a larger weight, and vice versa, dissimilar training data contribute less. To make a prediction on a data point (force on an atom in its local environment), the GP regression model requires the evaluation of the kernel between such point and all the N_T training set data points. The GP mean prediction is a weighted average of the training labels, using the distances as measured by the kernel function as weights. In Eq. (11), we notice that the predictive mean requires the evaluation of a $3N_T$-dimensional vector $\overline{\boldsymbol{k}}_{i\alpha}$, which quantitatively weighs the contribution to the test atomic environment ρ_i coming from the training atomic environments ρ_j.

In the following discussion, we consider a specific subset size n, capturing n-body atomic structure information, i.e., $(\overline{k}_{i\alpha}^{(n)})_{j\beta} = k_{\alpha\beta}^{(n)}(\rho_i^{(n)}, \rho_j^{(n)})$ without loss of generality. The same treatment below can be implemented on each n and summed over the mappings of all "n"s to get the final prediction if we are using a kernel function of the form of Eq. (5).

Starting with the idea introduced in ref. [34], the kernel definition (Eq. (6)) allows us to decompose the force prediction into contributions from n-atom subsets, and to decompose the variance into contributions from pairs of n-atom subsets.

To see this, we first note that the kernel vector of atomic environments ρ_i can be decomposed into the kernel of all the n-atom subsets in ρ_i, and the kernel of an n-atom subset p can be further decomposed into the contributions from all the $n - 1$ neighbor atoms (except for the center atom i). We denote $\mathbf{r}_p^s := \mathbf{R}^s - \mathbf{R}^i$ as a bond in p, where s labels the remaining $n - 1$ neighbor atoms, and introduce the unit vector $\hat{r}_p^s := \mathbf{r}_p^s / r_p^s$ denoting the direction of the bond. Then the force kernel has decomposition

$$\overline{\boldsymbol{k}}_{i\alpha}^{(n)} = \sum_{p \in \rho_i} \tilde{\boldsymbol{k}}_{i\alpha}^{(n)}(p) = \sum_p \sum_s \boldsymbol{\mu}_s^{(n)}(p) \hat{r}_{p,\alpha}^s , \tag{13}$$

where

$$\left[\tilde{k}^{(n)}_{i\alpha}(p)\right]_{j\beta} := \frac{\partial^2 \tilde{k}^{(n)}(p,q)}{\partial R^i_\alpha \partial R^j_\beta} \tag{14}$$

and

$$\left[\mu^{(n)}_s(p)\right]_{j\beta} := \sum_{q\in\rho_j}\sum_t \frac{\partial^2 \tilde{k}^{(n)}(p,q)}{\partial r^s_p \partial r^t_q}\hat{r}^t_{q,\beta} \tag{15}$$

are vectors in $\mathbb{R}^{3N_T}$ space. In the first equality of Eq. (13), we used the decomposition of Eq. (6), and in the second equality we first convert the derivative from atomic coordinates to the relative distances of the n-atom subset, then take advantage of the fact that the kernel only depends on relative distances between atoms, i.e., $r^s_p = \| \mathbf{r}^s_p \|$.

Therefore, forces may be written as

$$F^{(n)}_{i\alpha} = \sum_{p\in\rho_i}\sum_s f^{(n)}_s(p)\hat{r}^s_{p,\alpha} \ . \tag{16}$$

where

$$f^{(n)}_s(p) := \boldsymbol{\mu}^{(n)\top}_s(p)\boldsymbol{\eta} \tag{17}$$

and $\boldsymbol{\eta}$ is defined in Eq. (10).

As noted in ref. [34], we can achieve a better scaling by a computationally efficient parametric method, i.e., an approximate way for constructing such functions f. In fact, a parametric function can quickly yield the value without evaluating kernel distances from all the GP training points, making the cost independent of the size of the training set, thus remove a computational bottleneck of the full GP implementation.

In order to achieve an effective parametrization of these weight functions, we use a cubic spline interpolation on a uniform grid of points, which benefits from having an analytic form and a continuous second derivative. By increasing the number of points in the uniform grid, the MGP BFF prediction can be made arbitrarily close to the original functions.

The construction of the mapped force field includes the following steps:

Step 1: Define a uniform grid of points $\{x_k \in [a,b]^{\binom{n}{2}}\}$ in the space of distances given by the vectors $\mathbf{r}_p$. The lower bound $a \ge 0$ is chosen to be smaller than the minimal interatomic distance, and the upper bound b can be set to the cutoff radius of atomic neighborhoods, above which the prediction vanishes (using the zero prior of GP), and $\binom{n}{2} = \frac{n(n-1)}{2}$ is the number of interatomic distances in the n-atom subset.

Step 2: A GP model with a fixed training set yields the values $\{f^{(n)}_s(x_k)\}$ for each grid point x_k using Eq. (17).

Step 3: Interpolate with a cubic spline function $\hat{f}^{(n)}_s$ through the points $\{(x^{(n)}_k, f^{(n)}_s(x_k))\}$.

In prediction, contributions from all n-bodies are added up, so the force of local environment ρ is predicted as:

$$\hat{F}_\alpha(\rho) = \sum_n \sum_{p\in\rho^{(n)}}\sum_s \hat{f}^{(n)}_s(p)\hat{r}^s_{p,\alpha}.$$

Mapped variance field

Using a derivation similar to that of the force decomposition (Eq. (16)), we derive the decomposition of the variance

$$V_{i\alpha} = \sum_{n,n'} \sum_{\substack{p\in\rho^{(n)} \\ q\in\rho^{(n')}}} \sum_{s,t} v^{(n,n')}_{s,t}(p,q)\hat{r}^s_{p,\alpha}\hat{r}^t_{q,\alpha} \ , \tag{18}$$

where

$$v^{(n,n')}_{s,t}(p,q) := \delta_{nn'}\tilde{k}^{(n)}_{\alpha\alpha}(p,q) - \boldsymbol{\mu}^{(n)}_s(p)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^{(n')}_t(q) \ . \tag{19}$$

with the covariance matrix Σ defined in Eq. (9).

We note that the domain of the variance function v has dimensionality twice that of f, so that when the interaction order $n > 3$, the number of grid points becomes prohibitively large to efficiently map v. Even for a 3-body kernel, when v is a 6-D function, we find that the evaluation of the variance, both in terms of computational time and memory footprint, limits its usage to simple benchmarks.

An efficient evaluation of the variance is critical for adopting BAL. We now discuss a key result of this work and introduce an accurate yet efficient approximate mapping field for the variance. In particular, we focus on simplifying the vector $\boldsymbol{\mu}^{(n)}_s(p)$, which, from our tests, is the most computationally expensive term for predicting variance. In particular, since $\boldsymbol{\mu}^{(n)}_s(p)$ evaluates the kernel function between the test point p and all the training points, it scales linearly with the training set size. By implementing a mapping of the variance weight field, the cost of variance prediction depends only on the rank of our dimension reduction approach, and is independent of the training data set size.

To this aim, we first define the vector $\boldsymbol{\psi}^{(n)}_s(p) := \mathbf{L}^{-1}\boldsymbol{\mu}^{(n)}_s(p) \in \mathbb{R}^{3N_T}$, where $\mathbf{L}$ is the Cholesky decomposition of $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$, so that Eq. (19) can be written as:

$$v^{(n,n')}_{s,t}(p,q) = \delta_{nn'}\tilde{k}^{(n)}_{\alpha,\alpha}(p,q) - \boldsymbol{\psi}^{(n)}_s(p)^\top\boldsymbol{\psi}^{(n')}_t(q) \ . \tag{20}$$

The domain of vector function $\boldsymbol{\psi}^{(n)}_s(p)$ is half the dimensionality of that of $v^{(n,n')}_{s,t}(p,q)$.

As a first idea, since each component of $\boldsymbol{\psi}^{(n)}$ has domain of lower dimensionality, one may think of building $3N_T$ spline functions to interpolate the $3N_T$ components separately, which means the number of spline functions needed grows with the training set size.

To achieve a more efficient mapping of $\boldsymbol{\psi}^{(n)}_s$, we take advantage of the principal component analysis (PCA), a common dimensionality reduction method. The construction of the mapped variance field includes the following steps:

Step 1: We start by evaluating the function values on the uniform grid $\{x_k \in [a,b]^{\binom{n}{2}}, k = 1,\ldots,G\}$ as introduced in the section above. At each grid point x_k, we evaluate the vectors $\boldsymbol{\psi}^{(n)}_s(x_k)$ and use them to build the matrix $\mathbf{M}^{(n)}_s := (\boldsymbol{\psi}^{(n)}_s(x_1), \boldsymbol{\psi}^{(n)}_s(x_2), \ldots, \boldsymbol{\psi}^{(n)}_s(x_G))^\top \in \mathbb{R}^{G\times 3N_T}$.

Step 2: In PCA, we perform singular value decomposition on $\mathbf{M}^{(n)}_s$, such that $\mathbf{M}^{(n)}_s = \mathbf{U}^{(n)}_s\boldsymbol{\Lambda}^{(n)}_s(\mathbf{V}^{(n)}_s)^\top$, where $\mathbf{U}^{(n)}_s \in \mathbb{R}^{G\times 3N_T}$ and $\mathbf{V}^{(n)}_s \in \mathbb{R}^{3N_T\times 3N_T}$ have orthogonal columns, and $\boldsymbol{\Lambda}^{(n)}_s \in \mathbb{R}^{3N_T\times 3N_T}$ is diagonal.

Step 3: We reduce the dimensionality by picking the m largest eigenvalues of $\boldsymbol{\Lambda}^{(n)}_s$ and their corresponding eigenvectors from $\mathbf{U}^{(n)}_s$ and $\mathbf{V}^{(n)}_s$ (here we assume that eigenvalues are ordered). Then, we readily obtain a low-rank approximation as:

$$\mathbf{M}^{(n)}_s \approx \overline{\mathbf{M}}^{(n)}_s = \overline{\mathbf{U}}^{(n)}_s\overline{\boldsymbol{\Lambda}}^{(n)}_s(\overline{\mathbf{V}}^{(n)}_s)^\top \ , \tag{21}$$

with $\overline{\boldsymbol{\Lambda}}^{(n)}_s \in \mathbb{R}^{m\times m}$, $\overline{\mathbf{U}}^{(n)}_s \in \mathbb{R}^{G\times m}$, and $\overline{\mathbf{V}}^{(n)}_s \in \mathbb{R}^{3N_T\times m}$.

Step 4: Define $\boldsymbol{\phi}^{(n)}_s = (\mathbf{V}^{(n)}_s)^\top\boldsymbol{\psi}^{(n)}_s$, which is a vector in $\mathbb{R}^m$, and interpolate this vector between its values at the grid points $\{x_k\}$ (in detail, $\phi^{(n)}_{s,i}(x_k) = \overline{U}^{(n)}_{s,ki}\overline{\Lambda}^{(n)}_{s,ii}$, for $i = 1, \ldots, m$). As done previously for the force mapping, spline interpolation $\hat{\boldsymbol{\phi}}$ can now be used for approximating the $\boldsymbol{\phi}^{(n)}_s$ vector as well.

Therefore, the MGP prediction of the variance is given by:

$$\hat{v}^{(n,n')}_{s,t}(p,q) := \delta_{nn'}\tilde{k}^{(n)}_{\alpha,\alpha}(p,q) - \hat{\boldsymbol{\phi}}^{(n)}_s(p)^\top(\overline{\mathbf{V}}^{(n)}_s)^\top\overline{\mathbf{V}}^{(n')}_t\hat{\boldsymbol{\phi}}^{(n')}_t(q) \ . \tag{22}$$

In this form, the variance would still require a multiplication between two matrices of size $m\times 3N_T$. However, we note that the computational cost of total variance can be reduced

$$\hat{V}_\alpha(\rho) = k_{\alpha,\alpha}(\rho,\rho) - \left(\sum_n\sum_{p\in\rho^{(n)}}\sum_s \overline{\mathbf{V}}^{(n)}_s\hat{\boldsymbol{\phi}}^{(n)}_s(p)\hat{r}^s_{p,\alpha}\right)^2 \ . \tag{23}$$

Nominally, the calculation of the second term in the variance still scales with the data set size N_T, since the size of $\overline{\mathbf{V}}^{(n)}$ depends on N_T. However, we find the cost of the vector-matrix multiplication negligible compared to evaluations of the self-kernel function (the 1st term in Eq. (22)). Therefore, we have formulated a way to significantly speed up the calculation of the variance, which only weakly depends on the training set size. In addition, by tuning the PCA rank m, we can optimize the balance between accuracy and efficiency, increasing the flexibility of this model.

The convergence of the accuracy of mapped forces and uncertainty with different grid numbers G and ranks m is discussed in Supplementary Fig. 2.

DATA AVAILABILITY

The data and related code in this paper are published in Materials Cloud Archive[61] with https://doi.org/10.24435/materialscloud:qg-99[62].

CODE AVAILABILITY

The algorithm for the MGP and mapped Bayesian active learning has been implemented in the open-source package FLARE[16] and is publicly available online: https://github.com/mir-group/flare. The python scripts for generating the stanene force field, and MD simulations are also publicly available online: https://github.com/YuuuXie/Stanene_FLARE.

Received: 31 August 2020; Accepted: 16 February 2021;
Published online: 19 March 2021

REFERENCES

1. Van Duin, A. C., Dasgupta, S., Lorant, F. & Goddard, W. A. Reaxff: a reactive force field for hydrocarbons. *J. Phys. Chem. A* **105**, 9396–9409 (2001).
2. Chenoweth, K., Van Duin, A. C. & Goddard, W. A. Reaxff reactive force field for molecular dynamics simulations of hydrocarbon oxidation. *J. Phys. Chem. A* **112**, 1040–1053 (2008).
3. Baskes, M. Application of the embedded-atom method to covalent materials: a semiempirical potential for silicon. *Phys. Rev. Lett.* **59**, 2666 (1987).
4. Lindsay, L. & Broido, D. Optimized tersoff and brenner empirical potential parameters for lattice dynamics and phonon thermal transport in carbon nanotubes and graphene. *Phys. Rev. B* **81**, 205441 (2010).
5. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
6. Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **134**, 074106 (2011).
7. Behler, J. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **13**, 17930–17955 (2011).
8. Behler, J. Constructing high-dimensional neural network potentials: a tutorial review. *Int. J. Quantum Chem.* **115**, 1032–1050 (2015).
9. Behler, J. Perspective: machine learning potentials for atomistic simulations. *J. Chem. Phys.* **145**, 170901 (2016).
10. Mailoa, J. P. et al. A fast neural network approach for direct covariant forces prediction in complex multi-element extended systems. *Nat. Mach. Intell.* **1**, 471–479 (2019).
11. Smith, J. S., Isayev, O. & Roitberg, A. E. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
12. Batzner, S. et al. Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. https://arxiv.org/abs/2101.03164 (2021).
13. Shapeev, A. V. Moment tensor potentials: a class of systematically improvable interatomic potentials. *Multiscale Model. Simul.* **14**, 1153–1173 (2016).
14. Podryabinkin, E. V. & Shapeev, A. V. Active learning of linearly parametrized interatomic potentials. *Comput. Mater. Sci.* **140**, 171–180 (2017).
15. Hodapp, M. & Shapeev, A. In operando active learning of interatomic interaction during large-scale simulations. *Mach. Learn.: Sci. Technol.* **1**, 045005 (2020).
16. Vandermause, J. et al. On-the-fly active learning of interpretable bayesian force fields for atomistic rare events. *npj Comput. Mater.* **6**, 1–11 (2020).
17. Lim, J. S. et al. Evolution of metastable structures at bimetallic surfaces from microscopy and machine-learning molecular dynamics. *J. Am. Chem. Soc.* **142**, 15907–15916 (2020).
18. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104**, 136403 (2010).
19. Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
20. Bartók, A. P. & Csányi, G. Gaussian approximation potentials: a brief tutorial introduction. *Int. J. Quantum Chem.* **115**, 1051–1057 (2015).
21. Bartók, A. P., Kermode, J., Bernstein, N. & Csányi, G. Machine learning a general-purpose interatomic potential for silicon. *Phys. Rev. X* **8**, 041048 (2018).
22. Thompson, A. P., Swiler, L. P., Trott, C. R., Foiles, S. M. & Tucker, G. J. Snap: automated generation of quantum-accurate interatomic potentials. *J. Comput. Phys.* **285**, 316–330 (2015).
23. Schütt, K. et al. Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. In *Proc. Advances in Neural Information Processing Systems*, 991–1001 (2017).
24. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *J. Chem. Phys.* **148**, 241722 (2018).
25. Schütt, K. et al. Schnetpack: a deep learning toolbox for atomistic systems. *J. Chem. Theory Comput.* **15**, 448–455 (2018).
26. Zhang, L., Han, J., Wang, H., Car, R. & Weinan, E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **120**, 143001 (2018).
27. Wang, H., Zhang, L., Han, J. & Weinan, E. Deepmd-kit: a deep learning package for many-body potential energy representation and molecular dynamics. *Comput. Phys. Commun.* **228**, 178–184 (2018).
28. Han, J., Zhang, L., Car, R. & E, W. Deep potential: a general representation of a many-body potential energy surface. *Commun. Comput. Phys.* **23**, 629–639 (2018).
29. Musil, F., Willatt, M. J., Langovoy, M. A. & Ceriotti, M. Fast and accurate uncertainty estimation in chemical machine learning. *J. Chem. Theory Comput.* **15**, 906–915 (2019).
30. Peterson, A. A., Christensen, R. & Khorshidi, A. Addressing uncertainty in atomistic machine learning. *Phys. Chem. Chem. Phys.* **19**, 10978–10985 (2017).
31. Smith, J. S., Nebgen, B., Lubbers, N., Isayev, O. & Roitberg, A. E. Less is more: sampling chemical space with active learning. *J. Chem. Phys.* **148**, 241733 (2018).
32. Jinnouchi, R., Lahnsteiner, J., Karsai, F., Kresse, G. & Bokdam, M. Phase transitions of hybrid perovskites simulated by machine-learning force fields trained on the fly with bayesian inference. *Phys. Rev. Lett.* **122**, 225701 (2019).
33. Jinnouchi, R., Karsai, F. & Kresse, G. On-the-fly machine learning force field generation: application to melting points. *Phys. Rev. B* **100**, 014105 (2019).
34. Glielmo, A., Zeni, C. & De Vita, A. Efficient nonparametric n-body force fields from machine learning. *Phys. Rev. B* **97**, 184307 (2018).
35. Glielmo, A., Sollich, P. & De Vita, A. Accurate interatomic force fields via machine learning with covariant kernels. *Phys. Rev. B* **95**, 214302 (2017).
36. Glielmo, A., Zeni, C., Fekete, Á. & De Vita, A. Building nonparametric n-body force fields using gaussian process regression. in *Machine Learning Meets Quantum Physics*, 67–98 (Springer, 2020).
37. Chmiela, S., Sauceda, H. E., Müller, K.-R. & Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **9**, 1–10 (2018).
38. Zuo, Y. et al. Performance and cost assessment of machine learning interatomic potentials. *J. Phys. Chem. A* **124**, 731–745 (2020).
39. Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **117**, 1–19 (1995).
40. Tang, P. et al. Stable two-dimensional dumbbell stanene: a quantum spin hall insulator. *Phys. Rev. B* **90**, 121408 (2014).
41. Cherukara, M. J. et al. Ab initio-based bond order potential to investigate low thermal conductivity of stanene nanostructures. *J. Phys. Chem. Lett.* **7**, 3752–3759 (2016).
42. Bishop, C. M. *Pattern Recognition and Machine Learning* (Springer, 2006).
43. Murphy, K. P. *Machine Learning: a Probabilistic Perspective* (MIT Press, 2012).
44. Zhu, F.-f et al. Epitaxial growth of two-dimensional stanene. *Nat. Mater.* **14**, 1020 (2015).
45. Deng, J. et al. Epitaxial growth of ultraflat stanene with topological band inversion. *Nat. Mater.* **17**, 1081 (2018).
46. Liao, M. et al. Superconductivity in few-layer stanene. *Nat. Phys.* **14**, 344 (2018).
47. Ganz, E., Ganz, A. B., Yang, L.-M. & Dornfeld, M. The initial stages of melting of graphene between 4000 k and 6000 k. *Phys. Chem. Chem. Phys.* **19**, 3756–3762 (2017).
48. Los, J. H., Zakharchenko, K. V., Katsnelson, M. I. & Fasolino, A. Melting temperature of graphene. *Phys. Rev. B* **91**, 045415 (2015).
49. Zakharchenko, K. V., Fasolino, A., Los, J. & Katsnelson, M. Melting of graphene: from two to one dimension. *J. Phys. Condens. Matter* **23**, 202202 (2011).
50. Stukowski, A. Visualization and analysis of atomistic simulation data with OVITO-the Open Visualization Tool. *Modell. Simul. Mater. Sci. Eng.* **18**, 015012 (2009).
51. Khaliullin, R. Z., Eshet, H., Kühne, T. D., Behler, J. & Parrinello, M. Nucleation mechanism for the direct graphite-to-diamond phase transition. *Nat. Mater.* **10**, 693–697 (2011).
52. Bundy, F. Direct conversion of graphite to diamond in static pressure apparatus. *J. Chem. Phys.* **38**, 631–643 (1963).
53. Bundy, F. et al. The pressure-temperature phase and transformation diagram for carbon; updated through 1994. *Carbon* **34**, 141–153 (1996).
54. Irifune, T., Kurio, A., Sakamoto, S., Inoue, T. & Sumiya, H. Correction: ultrahard polycrystalline diamond from graphite. *Nature* **421**, 806–806 (2003).
55. Britun, V. F., Kurdyumov, A. V. & Petrusha, I. A. Diffusionless nucleation of lonsdaleite and diamond in hexagonal graphite under static compression. *Powder Metall. Met. Ceram.* **43**, 87–93 (2004).
56. Ohfuji, H. & Kuroki, K. Origin of unique microstructures in nano-polycrystalline diamond synthesized by direct conversion of graphite at static high pressure. *J. Mineral. Petrol. Sci.* **104**, 307–312 (2009).
57. Pang, W. et al. Epitaxial growth of honeycomb-like stanene on au (111). *Appl. Surf. Sci.* 146224 (2020).
58. Yuhara, J. et al. Large area planar stanene epitaxially grown on ag (111). *2D Mater.* **5**, 025002 (2018).
59. Gao, J., Zhang, G. & Zhang, Y.-W. Exploring ag (111) substrate for epitaxially growing monolayer stanene: a first-principles study. *Sci. Rep.* **6**, 1–8 (2016).
60. Rasmussen, C. E. Gaussian processes in machine learning. in *Summer School on Machine Learning*, 63–71 (Springer, 2003).

61. Talirz, L. et al. Materials cloud, a platform for open computational science. *Sci. Data* **7**, 1–12 (2020).
62. Xie, Y., Vandermause, J., Sun, L., Cepellotti, A. & Kozinsky, B. Fast bayesian force fields from active learning: study of inter-dimensional transformation of stanene. Materials Cloud Archive **2020.99** (2020) https://doi.org/10.24435/materialscloud:cs-tf.

ACKNOWLEDGEMENTS
We thank Steven Torrisi and Simon Batzner for help with the development of the FLARE code. We thank Anders Johansson for the development of the KOKKOS GPU acceleration of the MGP LAMMPS pair style. We thank Dr Nicola Molinari, Dr Aldo Glielmo, Dr Claudio Zeni, Dr Sebastian Matera, Dr Christoph Scheurer, Dr Nakib Protik, and Jenny Coulter for useful discussions. Y.X. is supported by the US Department of Energy (DOE) Office of Basic Energy Sciences under Award No. DE-SC0020128. L.S. is supported by the Integrated Mesoscale Architectures for Sustainable Catalysis (IMASC), an Energy Frontier Research Center funded by the US Department of Energy (DOE) Office of Basic Energy Sciences under Award No. DE-SC0012573. A.C. is supported by the Harvard Quantum Initiative. J.V. is supported by Robert Bosch LLC and the National Science Foundation (NSF), Office of Advanced Cyberinfrastructure, Award No. 2003725.

AUTHOR CONTRIBUTIONS
Y.X. developed the mapped variance method. J.V. led the development of the FLARE Bayesian active learning framework, with contributions from L.S. and Y.X. Y.X. implemented the Bayesian active learning training workflow based on the mapped Bayesian force field. L.S. and Y.X. developed the LAMMPS pair style for the mapped force fields. Y.X. trained the models for stanene and performed molecular dynamics and analysis. A.C. guided the setup of DFT calculations. B.K. initiated and supervised the work and contributed to algorithm development. Y.X. wrote the manuscript, with contributions from all authors.

COMPETING INTERESTS
The authors declare no competing interests.

ADDITIONAL INFORMATION
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-021-00510-y.

Correspondence and requests for materials should be addressed to Y.X. or B.K.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.