Application of Machine Learning-based Selective Sampling to determine $BaZrO_3$ Grain Boundary Structures

Tarjei Bondevik[1], Akihide Kuwabara[2], Ole Martin Løvvik[1,3]

[1]Centre for Materials Science and Nanotechnology, University of Oslo, Norway

[2]Japan Fine Ceramics Center, Nagoya, Japan

[3]SINTEF Materials Physics, Oslo, Norway

Abstract

A selective sampling procedure is applied to reduce the number of density functional theory calculations needed to find energetically favorable grain boundary structures. The procedure is based on a machine learning algorithm involving a *Gaussian process*, and uses statistical modelling to map the energies of the all grain boundaries. Using the procedure, energetically favorable grain boundaries in $BaZrO_3$ are identified with up to 85% lower computational cost than the brute force alternative of calculating all possible structures. Furthermore, our results suggest that using a grid size of 0.3 Å in each dimension is sufficient when creating grain boundary structures using such sampling procedures.

Introduction

Grain boundaries (GBs) play an important role in polycrystalline materials, as they affect macroscopic properties such as electrical conductivity, capacitance, and diffusion. In the proton-conducting oxide $BaZrO_3$, used as case material in this paper, resistive GBs lower the overall electrical conductivity significantly [1],[2],[3]. Studying GBs experimentally can be challenging, as the area of interest is in the range of a few nanometers. Thus, it is fruitful to also investigate GBs with a theoretical approach.

One option is to perform density functional theory (DFT) calculations on the grain boundaries, where a periodic supercell of the atomic GB structure is given as input, and the structure's calculated potential energy (PE) is the output. However, when constructing a GB supercell, we face a problem: the atomic GB structure is unknown and the number of possible configurations is large. Even for a simple coincidence-site lattice GB, one typically has to consider hundreds or thousands of possible configurations arising when shifting the two grains relative to each other in three dimensional rigid body translations [4].

To find the correct atomic GB structure, the brute force method is the most accurate: perform DFT calculations on all configurations, and define the GB structure as the structure with lowest PE. Due to the computational cost, only a limited number of GBs can be treated this way. As a GB increases in complexity, the number of atoms needed to describe its structure in a periodic supercell increases, rising the computational cost in two ways. First, the computational cost of a DFT calculation roughly goes as the cube of the number of atoms in the supercell. Second, with a larger GB supercell, the number of possible GB configurations increases, raising the number of required DFT calculations.

One alternative to DFT are classical interatomic potentials with significantly lower computational cost, such that complex GBs with large supercells can be described [5]. A problem with the classical interatomic potentials is however their limitations in accuracy, since the pair potentials usually have been developed for simpler geometries so that GBs are outside their validity range. This is increasingly problematic as the number of different elements increases, giving rise to an exploding number of many-body interactions.

To reduce computational cost while maintaining high accuracy, several reports in the literature apply machine learning-based techniques. Methods based on selective sampling with Bayesian optimization [6] has identified stable fcc-Cu [4] and metal oxide[7] GBs with a computational cost around two orders of magnitude lower than the brute force method. Other methods, where the algorithm was trained on a subset of GBs to predict properties of other GBs – sometimes labelled *transfer learning* – are able to effectively predict GB energies in fcc-Cu [8] and fcc-Al [9]. The combination of selective sampling and transfer learning is also fruitful, shown to reduce computational cost by around three orders of magnitude in determining the GB energies in Fe [10] and fcc-Al [11]. Furthermore, proton conduction has been characterized in $BaZrO_3$ [12] and atomic pathways has been identified in $BaZrO_3$ and $LaNbO_4$ [13], by combining DFT with selective sampling.

In this paper, we apply an approach inspired by the ones described in [4, 7, 12]:: combine DFT calculations with a machine learning-based selective sampling procedure, to identify the low PE grain boundary structures with as few DFT calculations as possible. A fine grid is introduced, where one grain is shifted relative to the other grain along the grid points. We define the potential energy surface (PES) to be the DFT calculated potential energy over the entire grid. Instead of calculating the entire PES, only a few grid points are calculated initially. Then we apply a statistical model that predicts the remaining part of the PES, based on the already calculated grid points. With a prediction of the entire PES, GB structures that are likely to have low PE can be identified. Then we apply selective sampling: the next grid point to calculate is selected based on the probability of that grid point being in a low PE region, as well as the uncertainty of that grid point. As more GB structures are calculated, the statistical model is updated and becomes increasingly accurate, particularly in the low PE region where most of the computational effort is focused. After a number of DFT calculations, all low PE structures are with a certain probability identified, and the sampling procedure can be terminated. At that point, only a few of the unphysical, high energy GB structures have been calculated. The accuracy from DFT calculations is maintained, but careful selective sampling prevents us from wasting computational resources on unphysical GBs.

$BaZrO_3$ is well-studied, and suitable as a case material in this study. Several DFT studies have been done on $BaZrO_3$ GBs, however, most of the studies either neglect the rigid body translations [14], [15], [16] or treat them inaccurately in a coarse grid [17]. Due to their relatively small supercell sizes, the $(210)[001]$ and $(111)[1\bar{1}0]$ GBs will be used for demonstration in the present work. For these particular GBs, calculating all possible grain shifts with the brute force method is possible with a reasonable computational cost. In more complex GBs however, the brute force method is ruled out due to large computational cost. Hence, this study serves as a demonstration of a method useful in more complex GBs.

Theoretical background

First, we discuss the number of possible GB configurations in greater detail, and how a grid with as few configurations as possible can be introduced. Since the grains can be shifted relative to each other in three dimensions, a large number of possible configurations arise. Second, we introduce the Gaussian Process (GP), the statistical model on which the procedure is based. In the GP, each grid point is assigned with an expected potential energy as well as an uncertainty of that energy. That enables us not only to identify configurations with low expected PE, but also configurations with high uncertainty, that is, areas where the GP model has low precision. Third, we explain the actual selective sampling procedure. After the model is initialized by calculating a few grid points, the next grid point to calculate is selected based on two considerations: the grid point should have a low expected PE, and a high expected uncertainty. Finally, the computational details of the DFT calculations are laid out.

Number of grain boundary configurations

In this consideration the (210) GB will be used as an example, but a similar procedure applies for the (111) GB. Detailed information about the GB supercells is shown in Table 1.

Table 1. Angles, dimensions, number of atoms, grain boundary separation of the GB supercells in this study. a_0 = 4.235 Å is the lattice parameter.

GB	Angles	Dimensions	# of atoms	GB separation [Å]
$(210)[001]$	$\alpha = 90°, \beta = 90°, \gamma = 90°$	$\sqrt{5}a_0 \times a_0 \times 4\sqrt{5}a_0$	100	19.54
$(111)[1\bar{1}0]$	$\alpha = 90°, \beta = 90°, \gamma = 60°$	$\sqrt{2}a_0 \times \sqrt{2}a_0 \times 4\sqrt{3}a_0$	60	14.82

With three geometrical degrees of freedom, the number of possible GB configurations is obviously very high, and we will reduce that number as much as possible. Keeping the right grain fixed, the left grain can be shifted in the *x*- and *y*-direction, shown in Figure 1 and Figure 2, respectively. Also, the GB will typically expand in the *z*-direction perpendicular to the GB plane: Due to a different chemical environment at the GB core than in the bulk, distances between the atomic planes tend to differ significantly. The two grains may also be twisted relative to each other. Since the supercell's periodic boundary conditions are violated when twisting the grains, a treatment within the DFT framework is non-trivial, and twisted GBs are therefore not considered in this work.

The geometrical freedom in the *z*-direction is trivial to eliminate. By only allowing atom movement and supercell expansion in the *z*-direction during DFT relaxation, the most energetically favorable GB expansion will be found and is hence not necessary to know prior to the calculation. Treating the degrees of freedom in the *x*- and *y*-direction is more complicated.

Applying a fine grid size of 0.12 Å in the *x*- and *y*-direction yields a mesh of $80 \times 36 = 2880$ grid points. Figure 1 shows the supercell with a view perpendicular to the *x*-direction. The left structure shows the symmetric tilt GB; in the right structure, Grain 1 is shifted 0.1 unit cell lengths in the *x*-direction relative to Grain 2. Increasing the grid size will reduce the number of grid points. However, in the *x*-direction, the atomic plane arrangement is complicated, and it is not straightforward to increase the grid size without risking missing the most energetically favorable GB configuration.

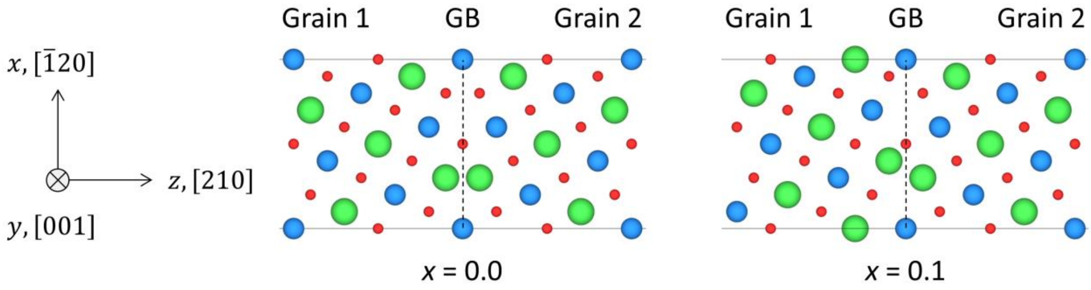

Figure 1. Supercells of the $(210)[001]$ grain boundary. In the left part, the grains are in the symmetric tilt configuration. In the right part, Grain 1 is shifted 0.1 unit cells in the *x*-direction, while Grain 2 is kept fixed. Only half of the atoms in the supercell are shown. Green, blue and red colors indicate Ba, Zr and O atoms, respectively. The black dashed line shows the grain boundary, the horizontal lines mark the supercell edges.

In the *y*-direction an increased grid size is possible, as shown in Figure 2. Due to symmetry, only configurations where *y* = 0 or *y* = 0.5 are likely, where *y* is the grain shift in direct coordinates, with the starting point being the symmetric tilt GB shown in Figure 2. In the case of a configuration with for example *y* = 0.3, the close-packed atomic planes in the *y*-direction are not aligned across the grain boundary. In a DFT calculation with no volume constraints, such a structure will therefore very likely relax into either a *y* = 0.0 or a *y* = 0.5 configuration. This was tested by relaxing several configurations with starting points $0.1 \leq y \leq 0.4$, and with no exceptions the configurations relaxed into either *y* = 0.0 or *y* = 0.5. Hence, the number of required grid points in the *y*-direction can be reduced to 2. Such reduction can, however, only be done in orientations with a special symmetry.

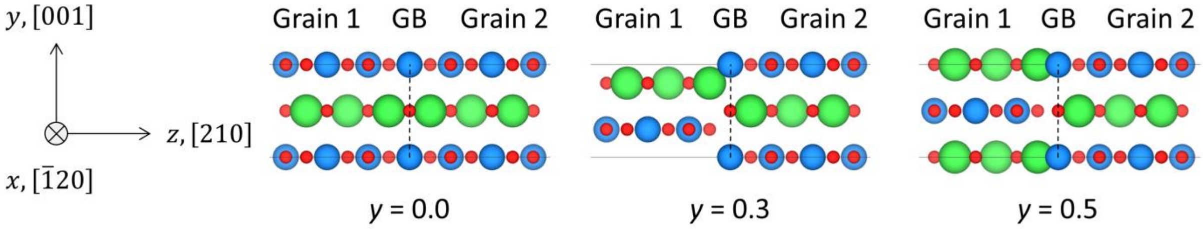

Figure 2. Supercells of the $(210)[001]$ grain boundary. In the left part (*y* = 0.0), the grains are in the symmetric tilt configuration. When the shift in the *y*-direction is 0.0 or 0.5, the atomic planes are aligned across the GB. However, with a shift between 0.0 and 0.5, the symmetry is lost; the structure is not stable and will, if allowed, relax into a y=0.0 or y=0.5 configuration during unconstrained relaxation. Less than half of the atoms in the supercell are shown. Colors as in Figure 1. The black dashed line shows the grain boundary, the horizontal lines mark the supercell edges.

The total number of grid points is then $80 \times 2 = 160$ for the (210) GB. Since we expect a low correlation in the DFT calculated energy between the $y = 0$ and $y = 0.5$ structures, it is convenient to treat these grid points as two separate datasets, shown as Datasets I and II in Table 2. In the (111) GB, the atomic plane arrangement is complex in both *x*- and *y*-directions, and it is not possible to reduce the grid density vastly in one of the directions. To keep the number of configurations reasonably low, a coarser grid of 0.50×0.50 Å^2 was applied, shown as Dataset III in Table 2.

Table 2. The datasets on which the selective sampling procedure are applied. In Datasets I and II, the shift in the *y*-direction is fixed, reducing the problem to one spatial dimension. In Dataset III, the shift in both *x*- and *y*-direction is free, leading to a two-dimensional problem.

Dataset	GB	*x*	*y*	Grid size	# of configurations
I	$(210)[001]$	Free	0.0	0.12 Å	80
II	$(210)[001]$	Free	0.5	0.12 Å	80
III	$(111)[1\bar{1}0]$	Free	Free	0.50×0.50 Å^2	144

The simple brute force method would be to calculate all configurations for all datasets, but that comes with a large computational cost. Rather, we will only calculate some configurations for each dataset, and use the Gaussian Process model to predict the entire PES.

Gaussian Process Model

In this section, we introduce the machine learning-based Gaussian Process (GP) model that describes the PES over the entire grid [18]. A GP model can roughly be seen as a collection of Gaussian functions: for the entire configuration space, the GP model returns the mean and the variance of the normal distribution at a given location. In our case, the GP model specifically returns the expected DFT calculated energy and its variance for each grid point: The potential energy E_i at grid point *i* is described as a normally distributed variable,

$$E_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right), \quad i = 1, \dots, N. \tag{1}$$

Here, *N* is the total number of GB configurations and $\mathcal{N}\left(\mu_i, \sigma_i^2\right)$ is the normal distribution of the potential energy at grid point *i*, with μ_i as the expected mean and σ_i^2 as the variance. The probability density function at each grid point *i* is then the usual $f_i(x) = \left(2\pi\sigma_i^2\right)^{-1/2} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma^2}\right)$.

A simple illustration of a one-dimensional GP model is shown in Figure 3, where the model is based on five observed data points. Close to observed data points (such as $x = x_2$) the model has a low uncertainty, while further away from the observed data points (such as $x = x_1$), the uncertainty increases.

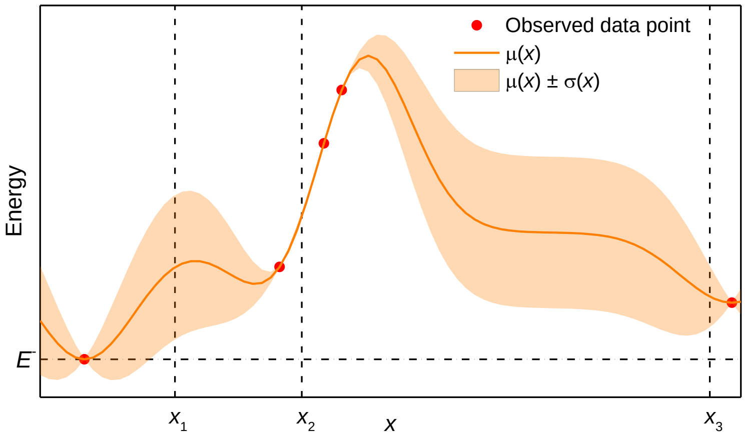

Figure 3. One dimensional GP model, with five observed (in this work: calculated by DFT) data points. The orange line shows the mean prediction, with the shaded area being the uncertainty. E^- is defined as the lowest energy among the observed data points.

Moving from one to *d* dimensions, we introduce a *d*-dimensional vector of descriptors as $\boldsymbol{\chi}_i \in \mathbb{R}^d$ for $i = 1, \dots, N$, where a descriptor can be interpreted as a variable describing the PES. The descriptors we use will be explained in a later section; an example of a descriptor is the grain shift in the *x*-direction. The covariance function *k*, which roughly describes the similarity between grid point *i* and grid point *j*, is defined as a radial basis function:

$$k(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) = \exp\left(-\frac{1}{2} \sum_{l=1}^{d} \left(\frac{\chi_{il} - \chi_{jl}}{\theta_l} \right)^2 \right). \tag{2}$$

Here, χ_{il} is the value of the *l*th descriptor at grid point *i*, and θ_l is the length scale parameter for the *l*th descriptor, automatically tuned by the "fmin_l_bfgs_b" algorithm from the Python `scipy.optimize` library [19]. The covariance function can be understood intuitively from Figure 3. There, only one descriptor is applied, meaning that $\boldsymbol{\chi}_i = x_i$. For two points *i* and *j* very close to each other, $|x_i - x_j| \approx 0 \implies k(x_i, x_j) \longrightarrow 1$, implying a high correlation between point *i* and *j*; for two points *i'* and *j'* far from each other, $k(x_{i'}, x_{j'}) \longrightarrow 0$, implying zero correlation.

To apply the GP model on the entire PES, we first need to calculate at least two potential energies to initiate the model. Calculating *t* potential energies from the grid, we define the *kernel matrix* as

$$\mathbf{K} = \begin{bmatrix} k(\boldsymbol{\chi}_1, \boldsymbol{\chi}_1) & \cdots & k(\boldsymbol{\chi}_1, \boldsymbol{\chi}_t) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{\chi}_\mathrm{t}, \boldsymbol{\chi}_1) & \cdots & k(\boldsymbol{\chi}_t, \boldsymbol{\chi}_t) \end{bmatrix}. \tag{3}$$

We apply the kernel matrix to provide a prediction of the mean and variance over the entire descriptor space [20]:

$$\mu(\boldsymbol{\chi}) = \mathbf{k}(\boldsymbol{\chi})^\mathrm{T} \mathbf{K}^{-1} \mathbf{E}, \tag{4}$$

$$\sigma^2(\boldsymbol{\chi}) = k(\boldsymbol{\chi}, \boldsymbol{\chi}) - \mathbf{k}(\boldsymbol{\chi})^\mathrm{T} \mathbf{K}^{-1} \mathbf{k}(\boldsymbol{\chi}). \tag{5}$$

Here, $\mathbf{E}$ represents a vector with the *t* calculated potential energies, $\mathbf{E} = [\mathrm{E}_1, \dots, \mathrm{E}_t]$, and $\mathbf{k}(\boldsymbol{\chi}) = [k(\boldsymbol{\chi}, \boldsymbol{\chi}_1), \dots, k(\boldsymbol{\chi}, \boldsymbol{\chi}_t)]^\mathrm{T}$. Each time we do a new DFT calculation, the GP model is updated with improved accuracy. To implement the model, the Gaussian Processes software from Python sklearn [21] is used, whose implementation is based on Algorithm 2.1 in [18].

Descriptors

In principle, any kind of descriptor can be used as input to the GP model. Yet, for a descriptor to be effective, it should be correlated with the output of the model, which in our case is the DFT calculated PES. In this work we have used five descriptors, shown in Figure 4.

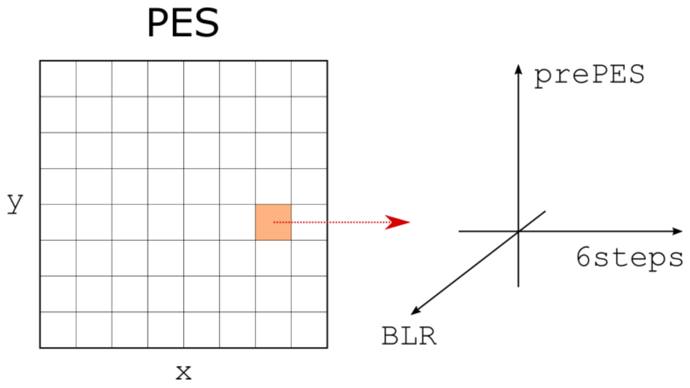

Figure 4. Relation between the PES and the other descriptors. Each spatial point in the PES is described by the *x* and *y* shift, as well as three additional descriptors. When all descriptors are included in the model, it becomes five dimensional.

The already mentioned `x` descriptor, and its equivalent `y`, are the two most obvious choices of descriptors, describing relative grain shifts in the *x*- and *y*-direction, respectively. It is reasonable to assume that two GB configurations close to each other in space have a similar potential energy. Hence, including these descriptors should improve the GP model.

There is also a correlation between the energy calculated from fast, inaccurate DFT calculations and slow, accurate ones. The `prePES` descriptor, standing for *preliminary* PES, exploits this principle. It describes the entire PES using deliberately sloppy convergence criteria (see Table 3 for details) with zero ionic relaxation steps, having around two orders of magnitude lower computational cost than a full DFT relaxation. Performing fast `prePES` calculations across the entire grid thus comes with a relatively low computational cost.

In the `6steps` descriptor, the DFT calculation is slow and accurate, and differs from the full calculation only at one point: the calculation is stopped after six ionic relaxation steps. The energies given by the descriptor will thus be more accurate than those from the `prePES` descriptor. However, acquiring all the data needed for the `6steps` descriptor comes with a significant computational cost, typically around one order of magnitude higher than the cost of acquiring the `prePES` descriptor. Hence, including the `6steps` descriptor will likely improve the GP model, but it is not clear if the total computational cost will be reduced.

The bond length ratio (`BLR`) descriptor captures deviations in the bond lengths of the GB supercell compared to bulk bond lengths. These bond length deviations correlate with higher calculated potential energies, indicating that this descriptor may improve the performance of the GP model. To calculate the `BLR` coefficient, let *i* be an atom type in the supercell that consists of *N* atoms. Each atom has three types of neighbors, with each neighbor type denoted as *j*, that is, $j = \{\mathrm{Ba}, \mathrm{Zr}, \mathrm{O}\}$. Each atom type *i* with neighbor type *j* has m_{ij} nearest neighbors. For example, in the $BaZrO_3$

perovskite structure, the Ba-Ba bond has $m_{ij} = 6$, and the Ba-O bond has $m_{ij} = 12$. The BLR coefficient is

$$\mathrm{BLR} = \frac{1}{2} \cdot \sum_{i}^{N} \sum_{j}^{3} \sum_{k}^{m_{ij}} \frac{|r_{ijk} - r_{bulk,ij}|}{r_{bulk,ij}}. \tag{6}$$

Here, the $\frac{1}{2}$ factor prevents double counting, r_{ijk} is the distance between the i^{th} and k^{th} atom and $r_{bulk,ij}$ the bulk distance between atom type *i* and atom type *j*. For example, for the Ba-O bond, $r_{bulk,ij} = a_0 \frac{\sqrt{2}}{2}$, where a_0 is the lattice parameter. Atoms located very close or very far from each other yields a high $|r_{ijk} - r_{bulk,ij}|$ value, increasing the BLR coefficient. Figure 5 provides an illustration of how the BLR coefficient is calculated.

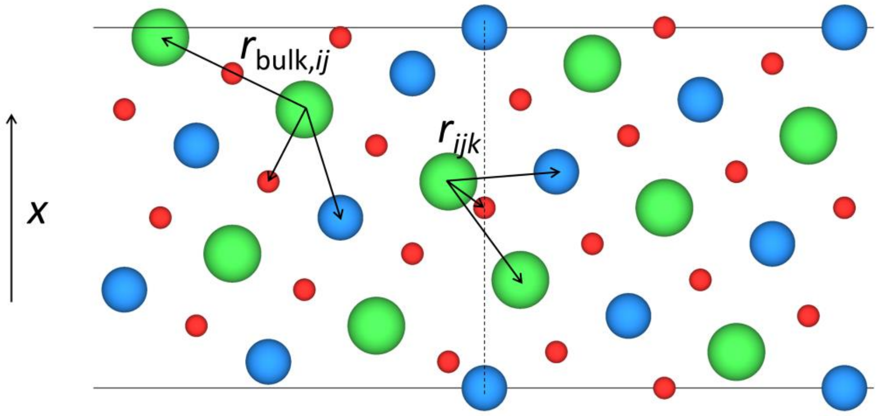

Figure 5. Illustration of the distances going into the BLR coefficient, for a (210) GB whose left grain is shifted 0.27 unit cell lengths in the *x*-direction. The interatomic distances differ significantly at the GB core compared to those at the bulk, illustrated by the length of the arrows. Colors as in Figure 1. The black dotted line shows the grain boundary, the horizontal lines mark the supercell edges.

The selective sampling procedure

The selective sampling procedure follows these steps:

a) Initialize the GP model by calculating N_{rand} random points from the grid
b) Select the next grid point to calculate
c) Calculate the next grid point, and update the GP model. Repeat b) and c) until convergence.

The initialization is straightforward: DFT calculations are performed on N_{rand} random grid points, and the resulting energies are used to initiate the GP model. To set N_{rand} one has to consider a trade-off between convergence rate and robustness: low N_{rand} values typically lead to faster convergence, but also a higher chance of convergence at a local rather than global minimum. In this work, $N_{\mathrm{rand}} = 10$ was found to be a suitable compromise.

When selecting the next grid point to calculate, we balance the tradeoff between exploration and exploitation. On one hand, we *exploit* the low PE region by spending most of our computational resources there. On the other hand, we also *explore* by calculating the high PE region with at least some precision, to be confident that we do not miss the global minimum.

To achieve a suitable balance, the *expected improvement* method from the Bayesian optimization literature was applied [20]. Specifically, we want to calculate the expected improvement relative to the minimum value of the *t* sampled data points, $E^{-} = \min_{i \in t}\{E_i\}$. The approach is to calculate the expected improvement for each grid point and select the "best" (here: the most negative) expected improvement for the next grid point to calculate. The expected improvement $\mathrm{EI}(\boldsymbol{\chi}_i)$ for grid point *i* is [20]

$$\mathrm{EI}(\boldsymbol{\chi}_i) = (\mu(\boldsymbol{\chi}_i) - E^{-})\big(1 - \Phi(Z_i)\big) - \sigma(\boldsymbol{\chi}_\mathrm{i})\phi(Z_i). \tag{7}$$

Here, $\Phi(Z_i)$ is the cumulative distribution function and $\phi(Z_i)$ the probability density function of the standard normal distribution, with *Z_i* being the usual standardized value given as $Z_i = \frac{\mu(\chi_i) - E^{-}}{\sigma(\chi_i)}$. Equation (7) reveals the balance between exploration and exploitation: the second term explores and may dominate when $\sigma(\boldsymbol{\chi}_\mathrm{i})$ is large; the first term exploits and usually dominates when $\mu(\boldsymbol{\chi}_i) < E^{-}$. Calculating the expected improvement for all grid points, its most negative value is selected as the next grid point i_{t+1} to calculate:

$$i_{t+1} := \arg\min_{i \in N} \mathrm{EI}(\boldsymbol{\chi}_i)\,. \tag{8}$$

After grid point i_{t+1} is calculated with DFT, the selection procedure is repeated until convergence. Our convergence criterion is based on the probability of improvement. As more grid points are calculated, the absolute value of the expected improvement will approach zero. This is because most or all grid points in the low PE region will already be calculated, and any improvement is unlikely. In this work, the sampling was terminated when $\min_{i \in N}|\mathrm{EI}(\boldsymbol{\chi}_i)| < 10^{-15}$, under the condition that $E_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$ was standardized when applied in the GP model. The standardization sets the mean to zero and the variance to unity, and is commonly used as preprocessing to enhance performance of machine learning algorithms. Our convergence criterion was compared to the false negative rate convergence criterion in [12], and they performed similarly.

Figure 6 provides an example of how the procedure works by showing the first eleven steps of a demonstration calculation where all values on a dense grid were calculated from before. For simple visualization, only the x descriptor is applied in a one-dimensional model. Both exploration and exploitation are balanced, step 6 being an example of the former, and step 4 an example of the latter. After 11 steps, it is clear that the selective sampling procedure focuses the computational effort on the low PE region. Since the goal of the procedure is to identify the lowest PE, this is an effective way to spend computational resources. For our purpose, there is nothing particularly interesting about the high PE region, other than knowing that it is in fact a high PE region. To be sufficiently confident of this, only a few calculations are needed there.

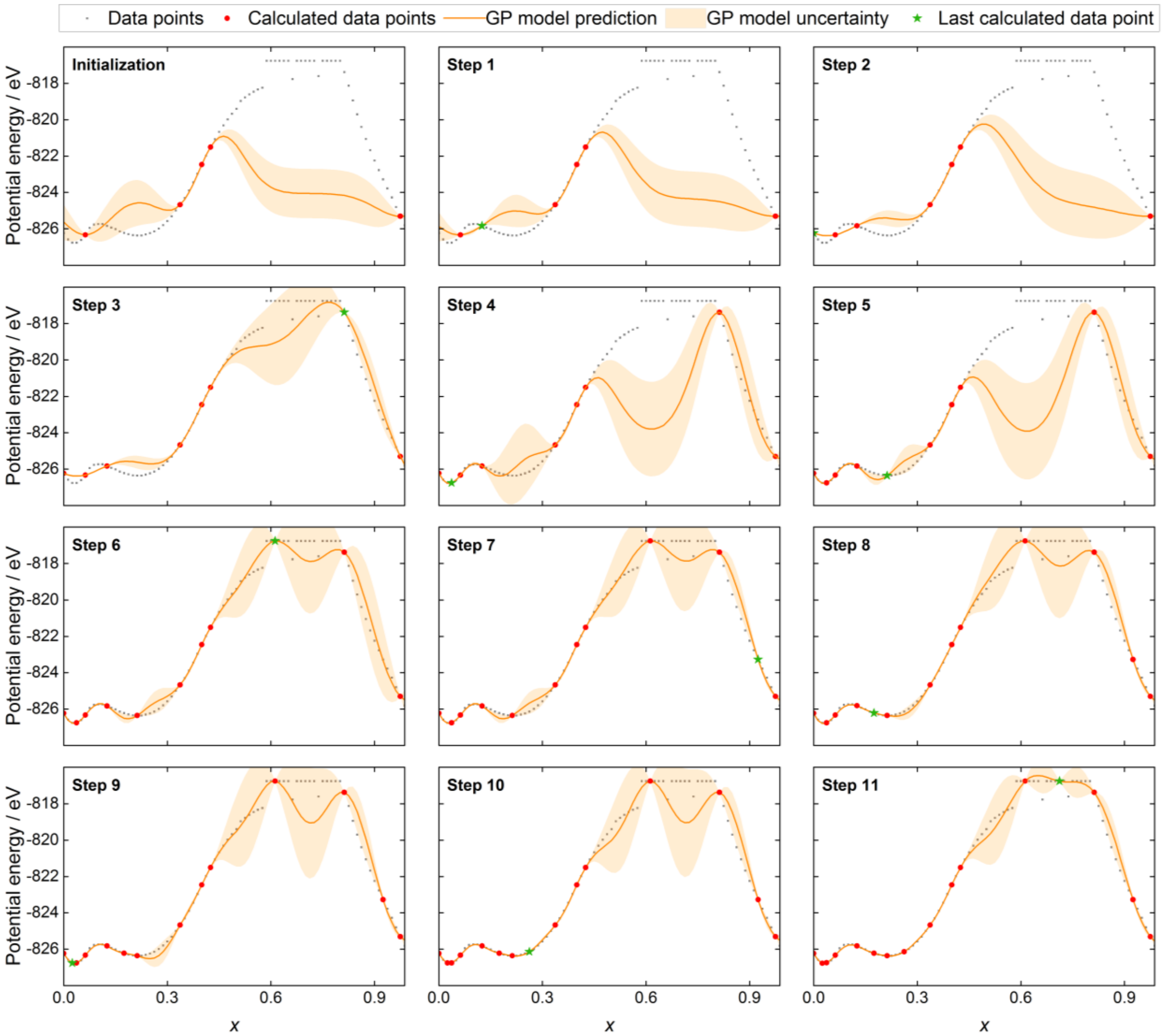

Figure 6. Example of how the GP algorithm identifies the lowest energy of a one-dimensional PES. *Data points* are the calculated potential energies from the entire grid; *Calculated data points* are the potential energies the GP model is based on; *GP model prediction* is the predicted expected energy given by equation (4); *GP model uncertainty* is the predicted standard deviation given by equation (5); *Last calculated data point* is the last selected data point based on the GP model from the previous step. For clearer visualization, five (not ten) initial random grid points are selected in this demonstration. A GP model based on those five initial points is made. Note that the GP model uncertainty is lowest close to the selected points. For steps 1-11, the algorithm selects the next grid point to calculate based on the expected improvement approach. In step 4, the lowest PE is actually identified, but we cannot know at that point. In step 6, one of

the highest PE are calculated, and after that, the algorithm can focus on other areas. Due to the applied energy filtering, all the highest PEs have the same constant value. The data in the figure are from Dataset I in Table 2.

Energy filtering

Sometimes the sampling procedure will select GB structures whose DFT calculations will either not converge or have a very large energy. These very large energies pose a problem for the algorithm, which is addressed by filtering out the largest energies. The energy filter can be seen in in Figure 6 for $0.59 < x < 0.78$. Defining ΔE_f as the energy filter, it is applied on the t calculated energies by

$$E_i := \arg\min\left\{\min_{i \in t}(E) + \Delta E_f, E_i\right\}. \tag{9}$$

With the energy filter, the largest and the non-converged energies are set to a fixed value, improving the GP model performance. By optimizing with respect to the convergence time of the sampling procedure, ΔE_f was set to 10 eV. The prePES and 6steps descriptors were filtered in a similar way. Since fast, inaccurate DFT calculations have a higher variance in the calculated energy, the magnitude of the energy filters must differ. By optimization, $\Delta E_f^{\mathrm{prePES}} = 3.0 \cdot \Delta E_f$ and $\Delta E_f^{\mathrm{6steps}} = 1.2 \cdot \Delta E_f$.

Computational details

To construct the grain boundary supercells, the Python package Atomic Simulation Environment (ASE) was applied [22]. The DFT calculations were performed using the projected augmented wave method [23] implemented in VASP [24, 25] employing the generalized gradient approximation [26], with key parameters shown in Table 3.

Table 3. Key parameters of the DFT calculations, where the plane wave cut off energies in the ordinary calculations had numerical precisions better than 0.5 meV for relative total electronic energies. In the ordinary calculations, *k*-point densities according to the Monkhorst-Pack scheme were applied; the prePES calculations were gamma-centered.

GB	Calculation	Plane wave cut-off energy / eV	k-point density	Convergence criterion, residual forces / eV Å^{-1}
(210)[001]	Ordinary	500	$3 \times 6 \times 1$[i]	0.02
(210)[001]	prePES	300	$2 \times 3 \times 1$	N/A
$(111)[1\bar{1}0]$	Ordinary	500	$5 \times 5 \times 1$	0.02
$(111)[1\bar{1}0]$	prePES	300	$2 \times 2 \times 1$	N/A

[i] The first relaxation was performed with the *k*-point density $2 \times 3 \times 1$ before $3 \times 6 \times 1$ was applied, to reduce the overall computational cost.

The GB supercells were relaxed in two steps. In the first step, the atoms and the supercell were only allowed to move and expand in the *z*-direction perpendicular to the GB. The selective sampling procedure was applied only on calculated potential energies from the first step. In the second step, we performed an unconstrained relaxation, where both the atoms and the supercell were allowed to move and expand in all directions. This unconstrained volume relaxation is computationally costly and was only performed on GB structures located at a local minimum after the first step. The GB structure with the lowest energy after the second step was then defined as the most energetically favorable structure.

Results and discussion

Lowest grain boundary energy

Table 4 shows the GB energy of the most energetically favorable configurations, found by unconstrained volume relaxations. The GB energy is defined as $\gamma_{\mathrm{gb}} = \left(E_{\mathrm{gb}} - E_{\mathrm{bulk}}\right)/2A$, where E_{gb} is the energy of the GB supercell, E_{bulk} the energy of the bulk supercell and *A* the GB interface area. Notably, the most energetically favorable (210) GB is the same GB that Lindman et al. identified when applying the brute force method on a coarse mesh of 5x2 grid points [17].

Table 4. The most energetically favourable grain boundary energies displayed for both GBs.

Grain boundary	*x* shift	*y* shift	$\gamma_{\mathrm{gb}}\ [\mathrm{J/m^2}]$
$(210)[001]$	32/80	1/2	1.13
$(111)[1\bar{1}0]$	0	0	0.51

Descriptor performance

To assess the descriptor performance, we first found the PES by performing DFT calculations on the entire grid. With a known PES, we could test if the selective sampling procedure indeed converged at the global energy minimum. This was done by simulating the sampling procedure with six different combinations of descriptors, on all three datasets. For each descriptor combination, the sampling procedure was simulated 500 times with different random seeds, to eliminate any effect from the random initialization. We assessed the descriptors by three defined parameters: the accuracy α, the fraction of the grid calculated at convergence β, and the fractional CPU cost. The accuracy is defined as

$$\alpha = \frac{n_{\mathrm{success}}}{n_{\mathrm{tot}}}, \tag{10}$$

where n_{success} is the number of simulations where the actual lowest PE was identified at convergence, and n_{tot} is the total number of simulations, set to 500 in this work. Occasionally, the sampling procedure converged at a local rather than global minimum, for example at the local minimum around $x = 0.2$ in Figure 6. Such deviation from optimal performance is captured by the α parameter. The fraction of the grid calculated at convergence is defined as

$$\beta = \frac{1}{n_{\mathrm{tot}}} \sum_{j=1}^{n_{\mathrm{tot}}} \frac{t_{\mathrm{conv},j}}{N}, \tag{11}$$

where $t_{\mathrm{conv},j}$ is the number of calculated grid points at convergence in the *j*-th simulation, and *N* again is the total number of grid points. Finally, the fractional CPU cost is defined as

$$CPU = \frac{1}{n_{\mathrm{tot}}} \sum_{j=1}^{n_{\mathrm{tot}}} \frac{\kappa_j}{\kappa_{\mathrm{tot}}}, \tag{12}$$

where κ_j is the CPU cost of the *j*-th simulation and κ_{tot} is the total CPU cost of calculating the entire grid with the brute force method. The goal is to achieve a high accuracy while calculating as few data points as possible, that is, maximize α, and minimize β and CPU.

Figure 7 shows how six different descriptor combinations performes on the (210) and the (111) GBs. Overall, the sampling procedure reduces the computational cost in the range of 65-85%, largely dependant on the descriptor combination in use. Notably, the procedure performs better on the (111) GB both in terms of accuracy and computational cost.

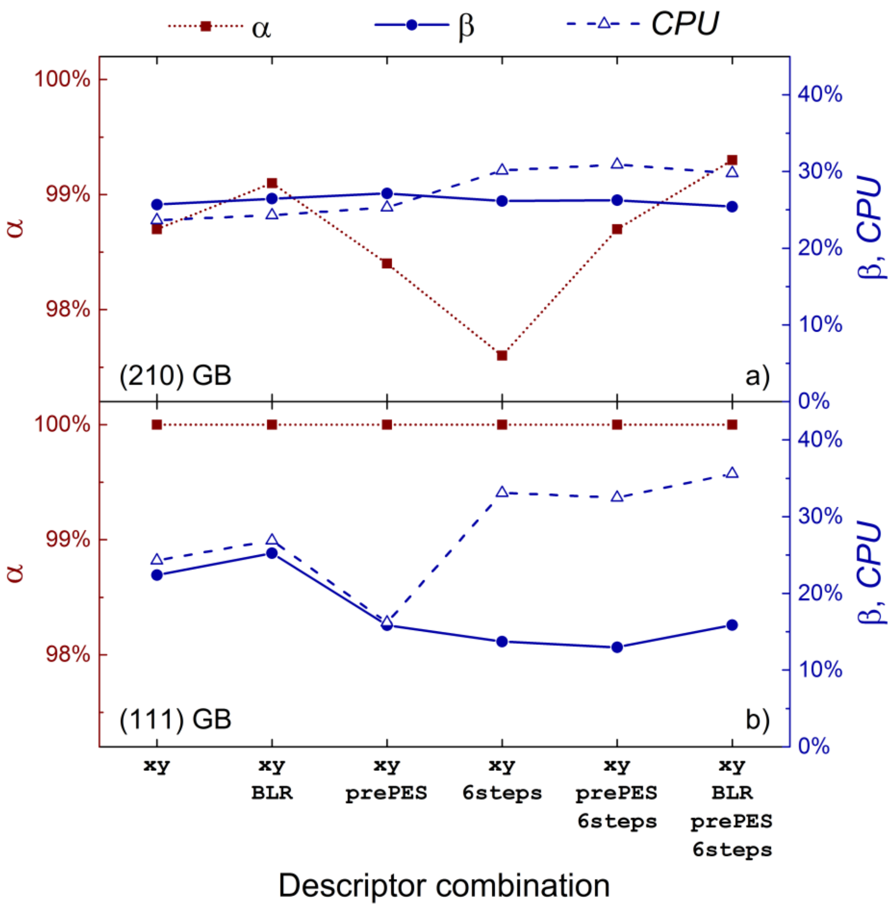

Figure 7. Performance of six different descriptor combinations, with α, β, and CPU defined in equations (10), (11), and (12). Each descriptor combination has been simulated 500 times for all three datasets. The results for the (210) GB (part a) are the average from Dataset I and II, with datasets being defined in Table 2. Note that since *y* is fixed in the (210) GB calculations, the y descriptor does not affect the results in part a).

Figure 7a) shows performance on the (210) GB, with the displayed data being the average from Datasets I and II. The results serve as an example of the challenge in understanding descriptor performance. Both the prePES and the 6steps descriptors lower the accuracy when applied only with the x descriptor. This is surprising in itself since these descriptors are highly correlated with the potential energy. When applying these descriptors simultaneously in the x+prePES+6steps combination, the accuracy somehow improves and is similar to the case of only applying the x descriptor. Choosing the optimal descriptor combination involves a trade-off. Applying all descriptors gives the highest accuracy (99.3%) but has also a relatively high CPU cost (29.8%). A better choice may be the x+BLR descriptor combination, with $\alpha = 99.1\%$ and $CPU = 24.3\%$.

In the (111) GB the picture is more clear and easier to explain intuitively, shown in Figure 7b). All descriptor combinations have 100% accuracy, meaning that the true global minimum was found in all simulations. The xy+6steps descriptor combination reached convergence with fewest steps.

However, the computational cost of acquiring the 6steps descriptor is high. The best performing descriptor combination in terms of computational cost is xy+prePES, with $CPU = 14.4\%$.

The 6steps descriptor performs differently on the two GBs. Although the complete causes of varying descriptor performance are hard to provide, it can partly be explained by differences in complexity between the two GBs. The (111) GB is simpler, with fewer ionic steps needed to obtain full relaxation. This means that relaxing with six ionic steps may yield a structure close to the fully relaxed structure. On the contrary, the (210) GB has a more complex structure, meaning that the structure obtained after the first six ionic steps may differ substantially from the fully relaxed structure. This difference becomes evident when considering the computational cost of acquiring the 6steps descriptor. In the (111) GB, the first six steps consumes on average 19.0 % of the total CPU cost from a full relaxation, compared to only 6.2 % for the (210) GB. The 6steps descriptor is hence expected to describe the PES better in the (111) GB than in the (210) GB, which normally would lead to a better performance in the (111) GB.

In both GBs, it is clear that adding more descriptors does not always help. The BLR descriptor performs poorly on the (111) GB, and adding it worsens the performance even in the case where BLR is combined with four other effective descriptors. Increasing the number of descriptors does not necessarily result in efficient prediction.

For future work on other GBs, one has to choose descriptors without prior knowledge of the energy landscape. Since the descriptors perform much differently on the two GBs this study is not clear on how to choose appropriate descriptors, but some careful recommendations can be made. The spatial `x` and `y` descriptors are certainly a good choice. In addition, the `prePES` descriptor seems to work well to balance accuracy with computational cost. It has by far the best performance in the (111) GB among the descriptors. The `6steps` descriptor is probably too costly to acquire, and although it performs well in the (111) GB, it has a surprisingly poor performance in the (210) GB. The `BLR` descriptor performs very differently on the two GBs, and more results from other GBs are needed to assess this descriptor. In summary, we recommend using the `x`, `y` and `prePES` descriptors for other GBs.

Finally, a note should be made about the convergence criterion. Here, there is an optimization trade-off: with a stricter criterion, all three parameters α, β and CPU will increase. Applying a convergence criterion lower than 1e-15 used in this work will push the accuracy in the (210) GB closer to 100%, but comes with a higher compuational cost.

Finding the optimal grid size

When choosing the grid size, there is a trade-off between accuracy and computational cost. A finer grid increases the probability of finding the global energy minimum, but also increases the computational cost. At one point, however, improving the grid resolution does not improve the accuracy any further. This is when two neighboring structures on the grid relax into the same structure during DFT relaxation. For example, as shown in Figure 2, the *y* = 0.3 shift for the (210) GB will relax into either *y* = 0.0 or *y* = 0.5, and hence a fine grid in the *y*-direction is not necessary.

To find the optimal grid size in the *x*-direction, we performed unconstrained volume relaxations on the entire Dataset I. By visually inspecting the 80 relaxed supercells, eight distinct structures were found. Quantitatively, the distinct structures can be identified by their relative grain shift and potential energy: if two relaxed GB structures are equivalent, their grain shift and their potential energy should be roughly similar. Figure 8a) shows the post-relaxation grain shift and potential energy as a function of pre-relaxation grain shift in the *x*-direction, where the eight distinct structures are marked with dotted lines. Figure 8b) visualizes these eight GB structures.

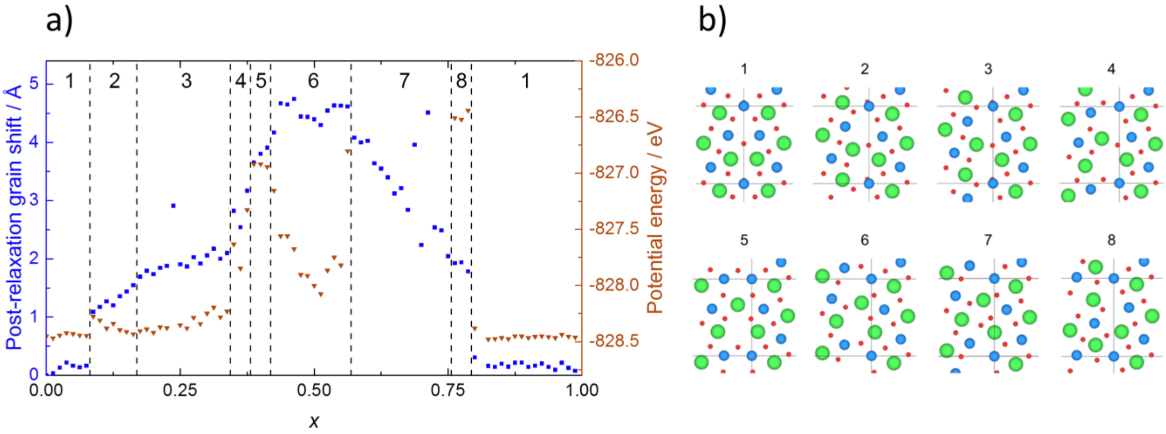

Figure 8. a) Post-relaxation grain shift and potential energy as a function of pre-relaxation grain shift in the *x*-direction. The post-relaxation grain shift is the *x* component-distance between to equivalent atoms in the middle of each grain. The DFT relaxations of the structures $0.575 \leq x \leq 0.75$ did not converge, and their potential energies are outside the range of this plot. b) Grain boundary structures, corresponding to the eight distinct structures in part a). The supercell edges are marked with horizontal lines; the grain boundaries are marked with a vertical line. Colors same as in Figure 1.

To find the optimal grid size, we observe that the narrowest distinct regions in Figure 8a) have a width of 0.36 Å, serving as an upper bound for the grid size. Based on this, we suggest that a grid size of 0.3 Å is sufficient to include all distinct GB configurations in the dataset. This yields a mesh of 32 x 2 grid points for the (210) GB, thereby reducing the computational cost of the brute force method with around 60%.

If a grid size of 0.3 Å is sufficient for other GBs than the (210) GB remains an open question. The optimal grid size should be related to density of chemical bonds that can form across the grain boundary, which in turn is related to the atomic density at the grain boundary. Although the various GBs have different atomic densities at the GB core plane, the densities are rather similar when also neighboring planes are included – and the neighboring planes do also contribute to chemical bonding across the grain boundary. Hence, we suggest that a grid size of 0.3 Å is generally reasonable for $BaZrO_3$ GBs.

A grid size of 0.3 Å implies a very large number of possible configurations for more complex GBs. For example, the $BaZrO_3$ (211) and (311) GBs, supercells will have 640 and 900 configurations, respectively. The computational cost of the brute force method is hence very large for these GBs and applying the proposed selective sampling procedure is probably a better choice.

Conclusions

1) The selective sampling procedure allows us to identify low energy grain boundary structures at a significantly reduced CPU cost. In the (210) GB, the CPU cost is reduced by around 75%

compared to the brute force method; in the (111) GB, the CPU cost is reduced by around 85%.

2) Adding sophisticated descriptors usually improves the accuracy of the procedure, but sometimes with a high computational cost. The `prePES` descriptor is the most promising descriptor as it may improve the convergence rate substantially at a low computational cost.
3) As a general rule, a grain shift grid size of 0.3 Å in each direction should be applied to ensure that the GB with lowest potential energy is found. For structures with a simpler symmetry in certain directions, such as the *y*-direction in the (210) GB, a coarser grid may be applied.

Acknowledgement

We would like to thank Kazuaki Toyoura and Espen Flage-Larsen for fruitful discussions. Computational resources have been provided by the Uninett Sigma2 National infrastructure for computational science in Norway. This work was supported by the “Functional oxides for clean energy technologies” (FOXCET) project, funded by the Research Council of Norway.

References

1. Kreuer, K.D., *Proton-conduction oxides.* Annual Review of Materials Research, 2003. **33**: p. 333-359.
2. Iguchi, F., et al., *The influence of grain structures on the electrical conductivity of a BaZr0.95Y0.05O3 proton conductor.* Solid State Ionics, 2006.
3. Bohn, H.G. and T. Schober, *Electrical Conductivity of the High-Temperature Proton Conductor BaZr0.9Y0.1O2.95.* J. Am. Ceram. Soc., 2000. **83**: p. 768–72.
4. Kiyohara, S., et al., *Acceleration of stable interface structure searching using a kriging approach.* Japanese Journal of Applied Physics, 2016.
5. Lindman, A., et al., *Oxygen vacancy segregation in grain boundaries of BaZrO3 using interatomic potentials.* Solid State Ionics, 2013.
6. Ueno, T., et al., *COMBO: An efficient Bayesian optimization library for materials science.* Materials Discovery, 2016. **4**: p. 18 - 21.
7. Kikuchi, S., et al., *Bayesian optimization for efficient determination of metal oxide grain boundary structures.* Physica B, 2018.
8. Kiyohara, S., et al., *Prediction of interface structures and energies via virtual screening.* Science Advances, 2016. **2**(11).
9. Tamura, T., et al., *Fast and scalable prediction of local energy at grain boundaries: machine-learning based modeling of first-principles calculations.* Modelling and Simulation in Materials Science and Engineering, 2017. **25**(7): p. 075003.
10. Oda, H., et al., *Transfer Learning to Accelerate Interface Structure Searches.* Journal of the Physical Society of Japan, 2017. **86**(12): p. 123601.
11. Yonezu, T., et al., *Knowledge-transfer-based cost-effective search for interface structures: A case study on fcc-Al [110] tilt grain boundary.* Phys. Rev. Materials, 2018. **2**: p. 113802.
12. Toyoura, K., et al., *Machine-learning-based selective sampling procedure for identifying the low-energy region in a potential energy surface: A case study on proton conduction in oxides.* Phys. Rev, 2016. **93**.
13. Kanamori, K., et al., *Exploring a potential energy surface by machine learning for characterizing atomic transport.* Physical Chemistry Chemical Physics, 2018.
14. Joakim Nyman, B., E.E. Helgee, and G. Wahnström, *Oxygen vacancy segregation and space-charge effects in grain boundaries of dry and hydrated BaZrO3.* Applied Physics Letters, 2012. **100**(6): p. 061903.
15. Polfus, J., et al., *Defect chemistry of a eBaZrO3 $ 3$ (111) grain boundary by first principle calculations and space-charge theory.* Physical Chemistry Chemical Physics, 2012. **14**: p. 12339-12346.
16. Helgee, E.E., A. Lindman, and m.G. Wahnstr\"o, *Origin of Space Charge in Grain Boundaries of Proton-Conducting eBaZrO3.* Fuel Cells, 2013. **13**: p. 19-28.
17. Lindman, A., E. Helgee, and m.G. Wahnstr\"o, *Theoretical modeling of defect segregation and space-charge formation in the eBaZrO3 (210)[001] tilt grain boundary.* Solid State Ionics, 2013. **252**: p. 121-125.
18. Rasmussen, C.E. and C.K.I. Williams, *Gaussian Processes for Machine Learning*, ed. M.I.T. Press. 2006: MIT Press.
19. Jones, E., et al., *SciPy: Open source scientific tools for Python*. 2001.
20. Brochu, E., V.M. Cora, and N. de Freitas, *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning.* arXiv:1012.2599, 2010.
21. Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 2011. **12**: p. 2825-2830.
22. Larsen, A.H., et al., *The atomic simulation environment—a Python library for working with atoms.* Journal of Physics: Condensed Matter, 2017. **29**(27): p. 273002.

23. Kresse, G. and D. Joubert, *From ultrasoft pseudopotentials to the projector augmented-wave method.* Phys. Rev. B, 1999. **59**: p. 1758-1775.
24. Kresse, G. and J. Hafner, *Ab initio molecular dynamics for liquid metals.* Phys. Rev. B, 1993. **47**: p. 558-561.
25. Kresse, G. and J. Furthm\"uller, *Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set.* Phys. Rev. B, 1996. **54**: p. 11169-11186.
26. Perdew, J.P., K. Burke, and M. Ernzerhof, *Generalized Gradient Approximation Made Simple.* Phys. Rev. Lett., 1996. **77**: p. 3865-3868.