Five Degree-of-Freedom Property Interpolation of Arbitrary Grain Boundaries via Voronoi Fundamental Zone Framework

Sterling G. Baird[a,*], Eric R. Homer[a], David T. Fullwood[a], Oliver K. Johnson[a]

[a]*Department of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA*

Abstract

In this work we introduce the Voronoi fundamental zone (VFZ) interpolation framework for grain boundary (GB) structure-property models and surrogates. The VFZ framework offers an advantage over other five degree-of-freedom (5DOF) based property interpolation methods because it is constructed as a point set in a manifold. This means that directly computed Euclidean distances approximate the original grain boundary octonion (GBO) distance with significantly reduced computation runtime ($\sim$7 CPU minutes vs. 153 CPU days for a $50\,000 \times 50\,000$ pairwise-distance matrix). This increased efficiency facilitates lower interpolation error through the use of significantly more input data. We demonstrate grain boundary energy (GBE) interpolation results for a non-smooth validation function and simulated bi-crystal datasets for Fe and Ni using four interpolation methods: barycentric interpolation, Gaussian process regression (GPR) or Kriging, inverse-distance weighting (IDW), and nearest neighbor (NN) interpolation. These are evaluated for $50\,000$ random input GBs and $10\,000$ random prediction GBs. The best performance was achieved with GPR, which resulted in a reduction of the root mean square error (RMSE) by 83.0% relative to RMSE of a constant, average model. Likewise, interpolation on a large, noisy, molecular statics (MS) Fe simulation dataset improves performance by 34.4% compared to 21.2% in prior work. Interpolation on a small, low-noise MS Ni simulation dataset is similar to interpolation results for the original GBO metric (57.6% vs. 56.4%). A vectorized, parallelized, MATLAB interpolation function (`interp5DOF.m`) and related routines are available in our VFZ repository (github.com/sgbaird-5dof/interp) which can be applied to other crystallographic point groups. The VFZ framework offers advantages for computing distances between GBs, estimating property values for arbitrary GBs, and modeling surrogates of computationally expensive 5DOF functions and simulations.

Keywords: Grain Boundary, Structure-Property Model, Interpolation, Octonion, Machine Learning

1. Introduction

1.1. Motivation

High fidelity GB structure-property models can accelerate the design and understanding of materials for GB engineering applications such as grain growth (GBE [1], mobility [2], and grain rotation [3–6]), stress-corrosion cracking (diffusivity [7, 8], solubility [9], and segregation [10]) [11–15], strength [16–18], ceramics [19, 20], electronics [21, 22], and thermoelectrics [23]. With the increased use of nanomaterials [16, 22], GBs take on increasingly larger roles as the GB volume fraction becomes significant; this is complicated by the fact that properties of GBs can span orders of magnitude depending on the five macroscopic degrees of freedom (DOFs) [24–26] as well as the three microscopic DOFs [27, 28]. However, the mentioned studies generally only consider a binary classification of GBs or variation of a few DOFs which

---

*Corresponding author.
   Email address:* `ster.g.baird@gmail.com` (Sterling G. Baird)

represents a small "slice" of the full grain boundary character (GBC) space. Recent advances in high-throughput simulation [26, 29–35], experimental characterization [26, 34, 36–38], and availability of rich GB datasets [33, 35, 39–47] warrant high-fidelity structure-property models capable of handling large amounts of input data to aid in the aforementioned applications.

1.2. Prior Work

In prior work, a number of strategies have been developed for predicting[1] 5DOF GB properties from experimental or simulated data. Because different works use different validation functions and data, it is difficult to objectively compare their performance. To facilitate meaningful comparisons, in addition to quoting absolute performance in terms of RMSE or mean absolute error (MAE), we will also report the percent reduction in error compared to a constant-valued control model whose value is chosen as the mean of the respective input data.

Several researchers have taken the approach of discretizing unsymmetrized 5DOF GBC space, and then using a least squares objective function and gradient descent to fit a piecewise-constant function, resulting in 5DOF grain boundary energy distributions for nickel [41], yttria [48], and copper [44] based on experimentally characterized 3D microstructures.

Restrepo et al. [49] used an artificial neural network (ANN) and approximately 17 000 and 51 000 Fe bicrystal simulations from Kim et al. [39] as training and validation data, respectively, to achieve MAEs of $0.0486\,\mathrm{J\,m^{-2}}$ and approximately $0.09\,\mathrm{J\,m^{-2}}$ in the best fitted ANNs for randomly selected and special GBs, respectively. If a constant, average value (i.e. average of the input GBEs) was chosen as the model, the MAE would be $0.0617\,\mathrm{J\,m^{-2}}$, implying that predictions of randomly selected GBs were improved by 21.2% relative to this simple, control model. Others have combined machine learning approaches with large

lists of macroscopic and microscopic descriptors [50, 51].

Recently, a new GB representation, GBOs, was reported [52] and tested [53]. The GBO representation is valuable for a number of applications. Most relevant to the present work is the resulting distance metric. The GBO distance metric offers an advantage over other metrics in that it "correctly determines the angular distances between GBs with a common normal or misorientation" and "closely approximates the geodesic metric on $SO(3)\times SO(3)$ *for all grain boundary pairs* while maintaining the ability to be analytically minimized with respect to the $U(1)$ symmetry" [52]. In this context, Francis et al. [52] derived octonion Spherical Linear Interpolation (oSLERP) and provided examples showing that oSLERP produces smooth, minimum distance paths through GB character space between two arbitrary GBs.

Laplacian kernel regression (similar to IDW) involving scaled pairwise distance matrices was later used with GBOs to predict properties of arbitrary GBs from a set of known values [53]. Using k-fold cross validation with $k = 10$ for 388 Ni GBE simulations [43] and an optimized scaling parameter, a RMSE of $0.0977\,\mathrm{J\,m^{-2}}$ was obtained compared to a constant, average model RMSE of $0.2243\,\mathrm{J\,m^{-2}}$ (56.4 % improvement). Due to computation time of pairwise distance matrices, this approach is currently "limited to datasets with several thousand or fewer" GBs [53].

1.3. Voronoi Fundamental Zone Framework

We present a new method for interpolating and predicting GB properties from a set of measured/calculated values (e.g. GBE from MS simulations). We term our approach the VFZ framework. It is highly efficient and facilitates the use of large data sets to enhance prediction accuracy. We discuss motivation for (Section 1.1) and prior implementations of 5DOF property prediction (Section 1.2) and then highlight unique properties of the VFZ framework that offer advantages over other methods (Section 1.3).

The VFZ interpolation framework introduced in this work offers an advantage over other methods

---

[1] We use the term "predict" throughout this work to refer to interpolation, inference, and/or extrapolation as some approaches can individually involve multiple prediction types.

because it is defined as a VFZ point set in a manifold[2] for which directly computed, scaled Euclidean distances approximate the original GBO distance given by Francis et al. [52]. This advantage is manifest in the ability to triangulate a mesh using standard routines (e.g. quickhull [55]) and interpolate using barycentric coordinates or machine learning methods such as GPR. Building on prior work on GBOs [52, 53], we create a VFZ point set by obtaining a set of GBOs minimized with respect to Euclidean distance and an arbitrary reference GBO after considering all symmetrically equivalent octonions (SEOs). Because GBOs are guaranteed to reside on the surface of a hypersphere [52] (a type of Riemannian manifold) a point set which locally resembles Euclidean space is the result (Section 2.1.3). Our implementation of the VFZ strategy utilizes GBOs; however, it may be possible to apply the same strategy using other distance metrics that have been proposed in the literature (for a comparison of many different metrics, see Morawiec [54]). Below we provide the detailed description of our methods (Section 2), followed by numerical test results (Section 3).

We also provide a vectorized, parallelized implementation of the VFZ framework and related functions. These are contained in what we will refer to as the VFZ repository, which is available at github.com/sgbaird-5dof/interp. In what follows, when we refer to built-in MATLAB functions, we refer to them with parentheses as in interp1(). When we refer to functions in the VFZ repository, we do so with the .m extension as in interp5DOF.m unless specifying the usage with arguments as in interp5DOF(qm,nA,qm2,nA2,y).

2. Methods

We describe methods related to the VFZ framework (Section 2.1.1), generation of random GBs (Section 2.2), and four different GB property interpolation schemes (Section 2.3). We also describe details regarding two simulated literature datasets that we use (Section 2.4).

2.1. The Voronoi Fundamental Zone Framework

The core operations of the VFZ framework are:

1. generating GBOs (Section 2.1.1)

2. mapping GBOs into a VFZ (Section 2.1.2)

3. calculating distances within the VFZ (Section 2.1.3)

2.1.1. Defining the Voronoi Fundamental Zone

Three degree-of-freedom fundamental zones (FZs) have typically been defined using linear inequalities (e.g. the orientation [56] and misorientation [56, 57] FZs). Instead of using linear inequalities[3], we take a numerical approach to define what we will call a VFZ.

To define a VFZ, an arbitrary, fixed, low-symmetry reference GBO is chosen (o_{ref}) and for our use of GBOs, the VFZ is defined as the region of $\mathbb{S}^7$ (the unit 7-sphere in 8 dimensions) closer to o_{ref} than any of its symmetric images[4]. However, use of the VFZ does not require its explicit construction. Rather, practical calculations require only the selection of the single point o_{ref} (which completes the definition of the VFZ), followed by mapping of query points into the VFZ by comparison of their SEOs with o_{ref}.

To illustrate the process of mapping points into the VFZ, we describe a 3D Cartesian analogue (Figure 1) to a 7D Cartesian non-degenerate (i.e. U(1) degeneracy removed) representation of a VFZ, which produces the standard stereographic triangle [58] for the O_h point group. A set of 500 points

---

[2]"In mathematics, a manifold is a topological space that locally resembles Euclidean space" [54]. By removing the Euclidean approximation in the VFZ framework and the renormalization of GBOs, the metric is intrinsic [54].

[3]If desired, linear inequalities can be obtained for a VFZ by determining a Voronoi tessellation's junction points (similar to what is shown in Figure 1 by e.g. voronoin()), transforming to 6D Cartesian coordinates via a singular value decomposition (SVD) transformation (Appendix B) and defining the bounded region by e.g. MATLAB FEX function vert2lcon.m.

[4]We also refer to lower-dimensional representations of the 8D Cartesian VFZ as VFZs (described in Section 2.1.2) and describe which dimensionality we are referring to as appropriate.

$(p_i, i \in [1, 500])$ randomly scattered on the surface of the 2-sphere comprise the data (red points in Figure 1a). A random point, p_{ref}, also on the surface of the 2-sphere, is chosen as the reference point (white circle). In this illustration, O_h or $m\bar{3}m$ point group rotations are used as symmetry operators, S_j, $j \in [1, N_p]$, where N_p is the twice the number of proper rotations due to inversion symmetry and $N_p = 48$ for the O_h point group. For each data point, 48 symmetrically equivalent representations $(p_{i,j}^{\text{sym}} = S_j(p_i)$, $j \in [1, 24])$ are produced by applying each of the relevant symmetry operators. After calculating the Euclidean distance between p_{ref} and $p_{i,j}^{\text{sym}}$, the point (p_i^*) closest to p_{ref} is chosen and retained as the unique representative of $p_{i,j}^{\text{sym}}$. As illustrated in Figure 1a, the projected points p_i^* (dark blue points) all fall in the VFZ without ever having to construct or define it explicitly. We call this group of projected points a *VFZ point set*. Note also that there is only one p_i^* in the VFZ for each $p_{i,j}^{\text{sym}}$ (see Figure 1b).

To calculate the distance between a given GBO, and the reference GBO, we employ the standard 8D Euclidean distance

$$d_{\text{E}}(o_A, o_B) = \left(\sum_{k=1}^{8} (o_{A,k} - o_{B,k})^2 \right)^{1/2} \qquad (1)$$

where $o_{A,k}$ and $o_{B,k}$ represent the k-the element of normalized GBOs o_A, and o_B, respectively.

Euclidean distance is an approximation to the true geodesic arc length on $\mathbb{S}^7$, which is given by

$$d_{\text{S}}(o_A, o_B) = \cos^{-1}(o_A \cdot o_B) \qquad (2)$$

where $\cdot$ is the dot product, $\cos^{-1}$ is the inverse cosine operator, and o_A and o_B are each normalized and $d_{\text{S}} \simeq d_{\text{E}}$ (Figure S1). In [52], the original GBO distance metric was defined by

$$d_{\Omega}(o_A, o_B) = 2\cos^{-1}(o_A \cdot o_B) \qquad (3)$$

where o_A and o_B are each normalized and d_{Ω} can be seen to be simply twice the geodesic arc length: $d_{\Omega} = 2d_{\text{S}}$. Thus, $d_{\text{E}} \simeq \frac{1}{2} d_{\Omega}$.

The definition of d_{Ω} has certain aesthetic benefits in that it mirrors the definition of a misori-

entation angle, ω_{AB}, between two crystal orientations in the quaternion parameterization: $\omega_{AB} = 2\cos^{-1}(q_A \cdot q_B)$.

Our choice to use d_{E} instead of d_{S} or d_{Ω} is motivated by the fact that it enables the use of standard algorithms, for a variety of operations, that require or assume Euclidean distances. In addition to enabling us to leverage the machinery of efficient and established algorithms, this choice can be justified by the following observations:

- The minimum Euclidean distance SEO will be the same as the minimum arc length distance SEO because d_{S} is a monotonically increasing function of d_{E}, for $d_{\text{S}}(d_{\text{E}}) \in [0, \pi]$ (Figure S1).

- For the FCC point group symmetry $(m\bar{3}m)$ the portion of $\mathbb{S}^7$ subtended by the VFZ is sufficiently small that the approximation $d_{\text{E}} \simeq d_{\text{S}}$ holds to very high accuracy[5] as shown in Figure S1.

- Calculation of d_{E} does not require the use of any inverse trigonometric functions and is about 23 % faster than calculation of d_{S} or d_{Ω}.

For applications other than interpolation which require precise quantification of high-dimensional volume, a mapping between Euclidean-approximated volumes and true volumes may be necessary[6] or the Euclidean approximation may be removed altogether[7]. The latter combined with removal of the renormalization of GBO allows for the (non-ensembled) VFZ-GBO metric to be intrinsic (see Morawiec [54] for an in-depth treatment of intrinsicality).

The expectation that a single, unique SEO will be found (within numerical tolerance and given

---

[5]This is true for a specific pair of GBOs within a VFZ. When calculating the *minimum* distance between SEOs of two points, there are additional considerations that must be attended to as discussed in detail in Section 2.1.3.

[6]We have not tested to what extent a Euclidean-approximated volume will differ from the true volume; however, Euclidean-approximated volumes can be obtained by using the triangulation methods discussed in Appendix B.1 (i.e. `convhulln()`.

[7]i.e. by setting the matlabdtype argument of `GBdist4.m` to `'omega'` rather than `'norm'`.
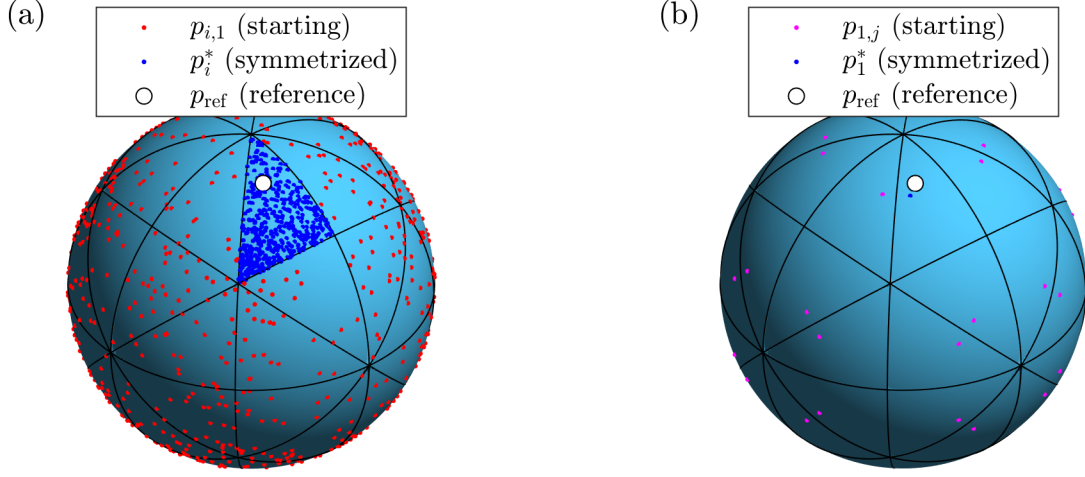
Figure 1: (a) 3D Cartesian analogue to a non-degenerate 7D Cartesian representation of U(1)-symmetrized GBOs and Voronoi fundamental zone grain boundary octonions (VFZ-GBOs) (VFZ-GBOs are inherently U(1)-symmetrized) which demonstrates the symmetrization of many points relative to a fixed reference point (white circle). This produces a 3D Cartesian VFZ point set (dark blue points). (b) To further illustrate, a single input point (magenta points) is symmetrized (dark blue point) relative to a fixed reference point (white circle), demonstrating that only one symmetrized point is found within the borders (black) of each of the Voronoi cells (light blue). The Voronoi tessellation is defined by the symmetric images of the reference point, and the spherical Voronoi diagram for this illustration is constructed using a modified version of [59].

a low-symmetry reference GBO[8]) is verified by several manual tests and internally within the symmetrization sub-routine `get_octpairs.m` [60] that is part of the `interp5DOF.m` package. Similar numerical tests reveal that inappropriately selecting a high-symmetry reference GBO to (attempt to) define a VFZ results in many degenerate minimum distance SEOs, with the identity ($\{1, 0, 0, 0, 0, 0, 0, 0\} \in \mathbb{R}^8$) [52] giving the highest degeneracy.

2.1.2. Mapping GBs to the Voronoi Fundamental Zone

As described above in the 3D analogy, with a reference GBO chosen (o_{ref}), and consequently the VFZ defined (Section 2.1.1), a GBO is mapped into the VFZ by finding among its SEOs the one that is closest to o_{ref} according to d_{E} (Eq. (1)). This is performed for all input and prediction points with respect to o_{ref}, and the result is a VFZ point set.

---

[8]The probability that a randomly generated GBO will fall exactly on a high-symmetry boundary vanishes in the limit of infinite precision.

2.1.3. Distance Calculations in the Voronoi Fundamental Zone

Euclidean distances are an accurate approximation of arc length distances in a VFZ because the difference between the two metrics for the maximum pairwise distance ($pd_{max} \simeq 60°$) in a VFZ is small as shown in Figure S1. However, when compared with the traditional GBO distance [52], due to the presence of low-symmetry GBs near the exterior of a VFZ, some GB pairs will exhibit larger Euclidean or arc length distances than is truly representative (see e.g. Figure 3a). In other words, moving "past" the low-symmetry border of a VFZ will result in an instantaneous relocation to a possibly distant point in the VFZ that in reality is highly correlated.

This is a limitation of the VFZ framework, which generates a VFZ with low-symmetry GBs at the borders in contrast to typical FZs [58, 61]. While defining a FZ with high-symmetry GBs at the borders (especially mirror-symmetry GBs) will certainly increase interpolation accuracy, the favorable interpolation results presented in this work are obtained because overestimation is infrequent within a small correlation length (e.g. 10° [42], which many NNs fall within for a 50 000 VFZ-
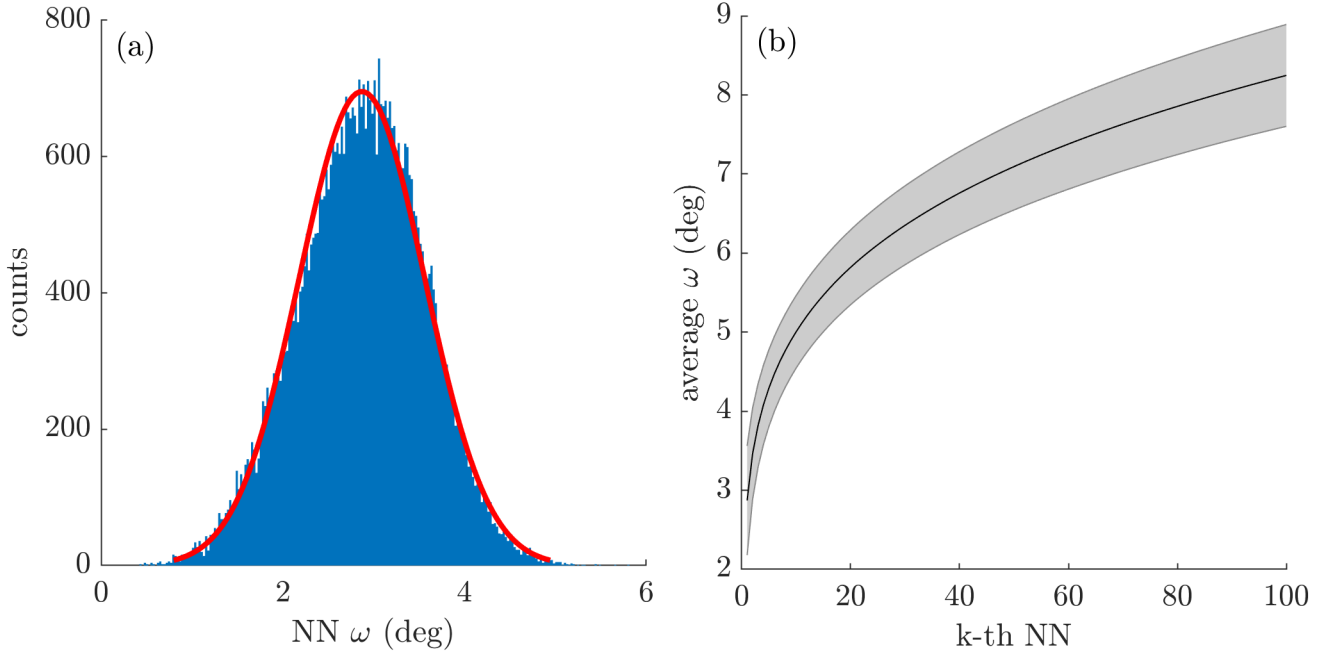
Figure 2: (a) Histogram of NN GBO distances (ω) in a VFZ-GBO set of 50 000 points. The average NN distance was $(2.870\,90 \pm 0.691\,12)°$. (b) The average k-th nearest neighbor distances demonstrate that many nearest neighbors fall within a tight tolerance (less then 10°) out of approximately 10 trial runs.

GBO set, see Figure 2b), and underestimation is non-existent within numerical precision. Naturally, smaller dataset pairwise distance matrices will exhibit more frequent distance overestimation.

Overestimation imposes a "sparseness" of data within a local region of influence common to the interpolation methods in this work, whereas underestimation would give erroneous high correlations between uncorrelated GBs. Because only overestimation relative to traditional GBO distances exist in this work (as shown in Figure 3), we expect that large errors will occur infrequently (Section 3.1).

While distance calculations are subject to these infrequent overestimates, they are largely immaterial for interpolation. This is because all interpolation methods in this work involve a region of influence that is small, so that if the distance to a NN is overestimated it simply does not contribute to the interpolation (the "sparseness" referred to earlier). Consequently the accuracy of the interpolation is not significantly impacted by infrequent distance overestimates, and excellent results can be achieved without addressing this limitation. However, if even greater accuracy is desired it can be

obtained for a relatively minor cost by considering multiple VFZs.

We find that taking the minimum distance among several VFZ-GBO sets defined by separate reference GBOs leads to better correlation between the Euclidean approximation and the traditional GBO metric as shown in Figure 3. Additionally, Figure 4 shows that the error between scaled Euclidean distance and the traditional GBO metric decreases rapidly as the number of ensemble VFZ-GBO components increases. This confirms that employing a small ensemble of VFZ-GBO sets results in significant improvement to the Euclidean distance approximation (Figures 3 and 4) of the traditional GBO metric. However, as already mentioned, improvements to interpolation results are expected to be less significant since they are already robust to occasional distance overestimates. In terms of computational runtime, use of an ensemble of 10 VFZs will increase runtime by a factor of $\sim$10 via a loop-based implementation. For a symmetrized 50 000 $\times$ 50 000 pairwise distance matrix, this results in a runtime of approximately 1 CPU hour instead of $\sim$7 CPU minutes for a sin-
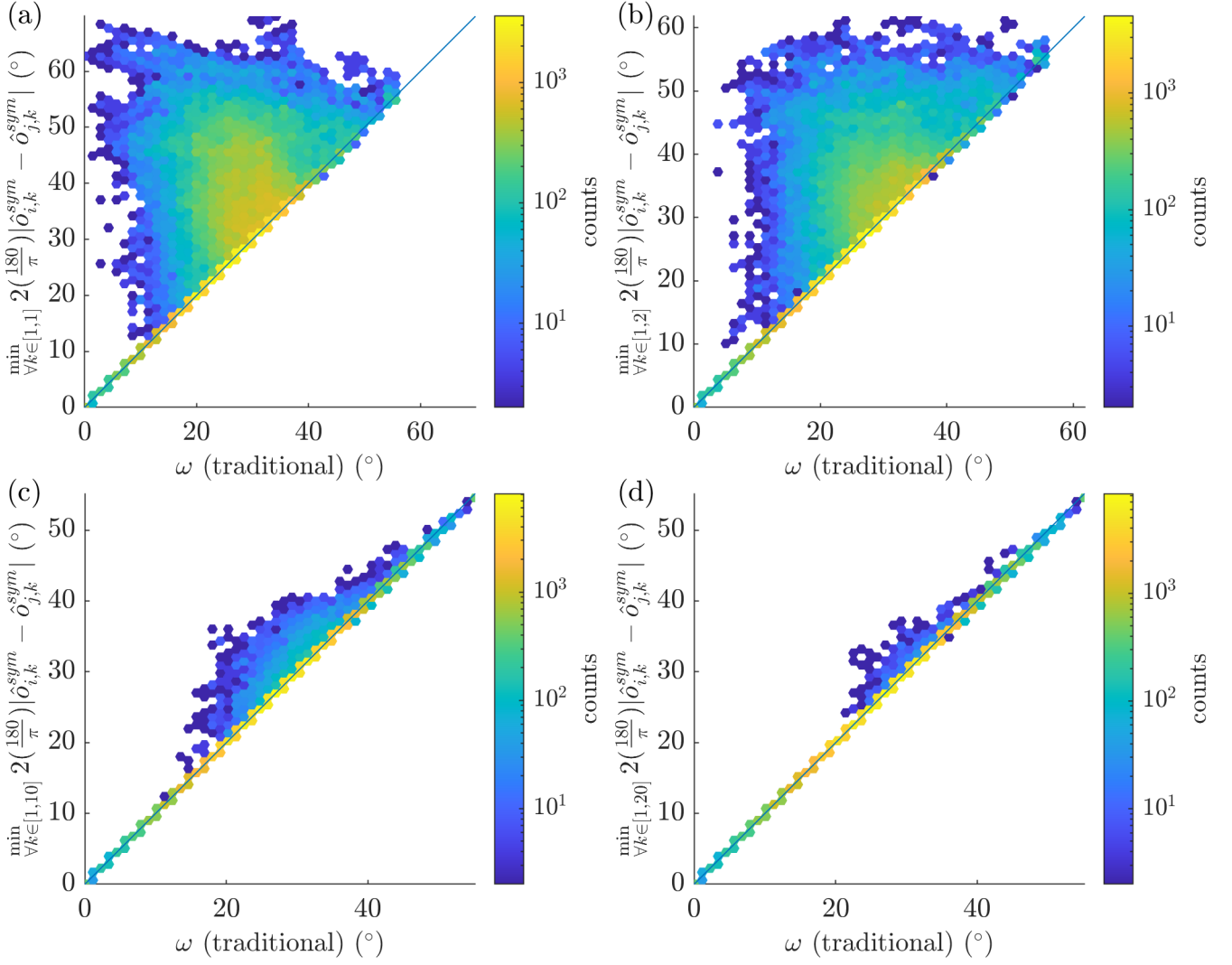
Figure 3: Hexagonally binned parity plots of pairwise distances of 388 Ni bicrystals [43]. Euclidean distance approximation is converted to GBOs ($x_{i,j,k} = 2\left(\frac{180}{\pi}\right)|\hat{o}_{i,k}^{\mathrm{sym}} - \hat{o}_{j,k}^{\mathrm{sym}}|$) for comparison with the traditional GBO metric [53]. The minimum distance among an ensemble of VFZ-GBO sets ($\min_{\forall k \in [1,k_{max}]} x_{i,j,k}$) is used for (a) 1, (b) 2, (c) 10, and (d) 20 VFZ-GBO sets. As the number of VFZ-GBO sets increases, the correlation between the Euclidean distance and the traditional GBO distance improves.

gle VFZ. However, this is still much faster than the original GBO approach used in [53], which would take an estimated 6.6 CPU years using the original implementation (or 153 CPU days if one GB in the GB pair is fixed according to the assumption in Morawiec [54]). Additionally, it may be worthwhile to make the distance calculations GPU-compatible for further speed-up.

VFZ-GBO Euclidean, hyperspherical arc length, and GBO distances are computed via VFZ repository function `GBdist4.m` which is used in the symmetrization function `get_octpairs.m` and an ex-

ample of ensemble VFZ-GBO distance calculations is given in `plotting.m`.

In addition to their use for distance calculations alone, ensembles of VFZ-GBO sets can be employed with interpolation methods to increase overall interpolation accuracy, but there is a computational cost (e.g. approximately $10\times$ using an ensemble of 10 VFZ-GBO sets). For $50\,000$ input points, use of an ensemble with 10 VFZ-GBO sets decreases RMSE and MAE from $0.0241\,\mathrm{J\,m^{-2}}$ and $0.0160\,\mathrm{J\,m^{-2}}$ to $0.0187\,\mathrm{J\,m^{-2}}$ and $0.0116\,\mathrm{J\,m^{-2}}$, respectively (single trial run). We expect these over-
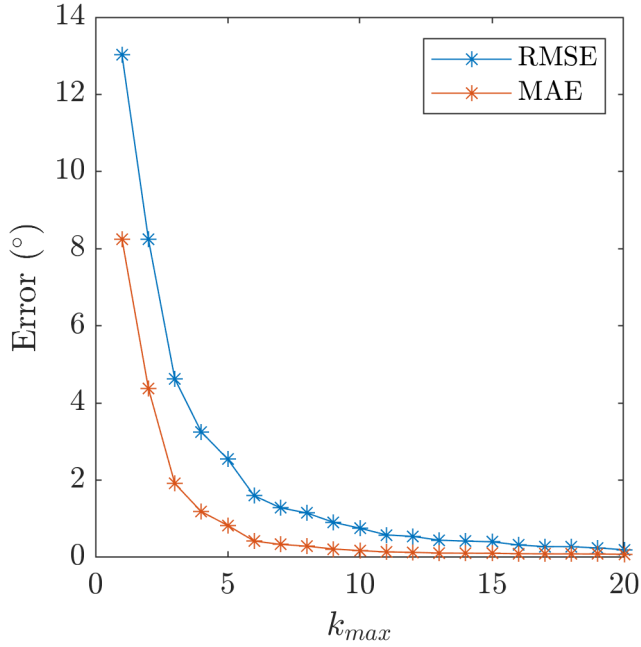
Figure 4: RMSE and MAE of pairwise distance errors for 388 Ni bicrystals [42] of scaled Euclidean distance approximation relative to the traditional GBO metric [53] (compare with Figure 3). The minimum distance among an ensemble of VFZ-GBO sets ($\min_{\forall k \in [1, k_{max}]} x_{i,j,k}$, where $x_{i,j,k}$ is the scaled Euclidean distance) is taken, iteratively adding consecutive sets up to $k_{max} = 20$. As the number of VFZ-GBO sets increases, RMSE and MAE between the scaled Euclidean distance approximation and the traditional GBO distance decreases.

all accuracy improvements occur because GBE predictions near the exterior of the VFZ where data may be sparse are improved. Ensemble interpolation results as a function of ensemble size and parity plots for mean, median, minimum, and maximum functions applied to the ensemble are shown in Figure S5 and Figure S6, respectively. Further details of ensemble interpolation are given in Section S3.

2.1.4. Comparison with Traditional Octonion Framework

We compare the VFZ framework with the traditional GBO metric (Table 1) and give examples that illustrate the computational complexity of each approach.

The construction of the VFZ dramatically reduces the computational burden of pairwise distance calculations. The mechanism by which this reduction is achieved can be illustrated with an ex-

ample. Let o_1 and o_2 denote two GBs represented in GBO coordinates. To perform a traditional symmetrized GBO distance calculation according to Francis et al. [52], we compare all SEOs of o_1 to all of the SEOs of o_2 and take the smallest distance. If N_p is the number of proper rotations of the crystallographic point group, this single minimum distance calculation requires a total of $4N_p^4$ SEOs to be considered (Sections 4.3 and 4.5 of Francis et al. [52]). Thus, the total number of SEO computations will be $4N_p^4 L^2$. However, it is possible to fix a single GB in the GB pair and still obtain accurate[9] due to isometry equivalence (see Section 7 of [54] and Figure S2).

In contrast, for a single distance calculation using the VFZ framework, o_1 and o_2 are first mapped into the VFZ, and then only a single distance calculation is required between them. Mapping o_1 into the VFZ requires comparison of $8N_p^2$ SEOs[10] of o_1 with a fixed reference GB in the interior of the VFZ; and likewise for o_2. Consequently, a single distance calculation between o_1 and o_2 under the VFZ framework requires $O(N_p^2)$ SEO computations. If one desires to compute a pairwise distance matrix between L GBs, the total computational cost[11] will be $O(N_p^2 L)$, which represents a dramatic reduction compared to the traditional approach. A summary of the differences between the two approaches is provided in Table 1.

2.2. Generating Random Voronoi Fundamental Zone Grain Boundary Octonions

In addition to the 3 core operations of the VFZ framework described in Section 2.1, it will be necessary for our tests, and useful for other applications,

---

[9]Compared with the pairwise distance matrix of the 388 Olmsted GBs, we obtained a RMSE of 1.6566×10^{-7}° for this computation which completed in 133 s using 6 cores (see `get_pd_fix.m`)

[10]This is 8 instead of 4 because the simplifying assumption that only two of the four double cover cases need to be considered [52] does not apply in the VFZ framework. This is confirmed by applying `uniquetol()` on a set of 4608 GBOs which has a final set size of 4608, where $4608 = 8 \times N_p^2$ and $N_p = 24$ (see `osymset.m`).

[11]See Section 3.2.2 for a detailed explanation of why this is *not* $O(N_p^2 L^2)$.

Table 1: Comparison between Voronoi fundamental zone grain boundary octonion and traditional GBO frameworks. *6D Cartesian representation used only for mesh triangulation efficiency in barycentric interpolation and *7D Cartesian representation only required for barycentric interpolation. 7D Cartesian representation is also implemented (though not required) for GPR, NN, and IDW. For pairwise distance complexity, N_p is the number of proper rotations ($N_p = 24$ for $m\bar{3}m$ face-centered cubic point group) and L is the number of GBs.

Property	Traditional	This Work
Symmetrizing Distance	GBO	VFZ Euclidean
Dimensionality	8D Cartesian	6*/7*/8D Cartesian
Bounded by FZ	No	Yes
Pairwise Distance Complexity	$O(N_p^2 L^2)$	$O(N_p^2 L)$
Rotation Convention	Passive	Active

to be able to generate random GBOs from 5DOF representations. We briefly explain here our process for accomplishing this.

First, random GBOs are formed by taking random misorientation quaternion (`qm`) and boundary plane normal (`nA`) pairs. Random misorientation quaternions are obtained via cubochoric sampling [62] (`get_cubo.m`) and random boundary plane vectors are sampled from a multivariate Gaussian distribution ($\mu = 0$, $\sigma = 1$) in $\mathbb{R}^3$ and normalized[12]. After this, they are converted to GBOs via VFZ repository function `five2oct.m`. The VFZ repository function `get_five.m` returns the result of these several operations. These (`qm`,`nA`) pairs are then converted to an GBO representation, `o`, using VFZ repository function `o=five2oct(qm,nA)` (see also VFZ repository function `get_ocubo.m` for generating random GBOs directly).

The GBOs are then symmetrized (i.e. they become VFZ-GBOs) via `osym=get_octpairs(o)`. A default reference GBO[13] is used for these calculations, unless specified by the user. We use the ac-

tive convention for `qm`, `nA`, and `o` (see Appendix A for further details of conventions).

For the present work we use this procedure to randomly generate VFZ-GBO sets containing between 100 to 50 000 VFZ-GBOs where each trial run has its own unique set of GBs. We use these to perform the validation and performance evaluation tests described later. For reference, we note that the average NN distance (over approximately 70 trials) of such sets ranges between $(10.7175 \pm 0.3684)°$ and $(2.6479 \pm 0.2254)°$, respectively.

Figure 5 illustrates how the VFZ-GBO average NN distance varies with the cardinality of the set (i.e. number of random VFZ-GBOs in the set). For a specific 50 000 VFZ-GBO set, the NN GBO distance is $(2.870\,90 \pm 0.691\,12)°$ (Figure 2a) while the average 100-th NN distance is within 10° (Figure 2b). This indicates that, on average, prediction VFZ-GBOs fall within a typical GB correlation length (10° [42]) of input VFZ-GBOs in large set sizes.

2.3. Interpolation in the Voronoi Fundamental Zone Framework

With the VFZ framework established, it is possible to define interpolation schemes over the VFZ to predict the properties of new GBs from the known properties of other GBs. For one application of interest to us, it is necessary to evaluate multiple different functions over a fixed set of input and prediction GBOs. In this section we first present a barycentric interpolation method that we

---

[12]Several methods for uniform sampling of points on a sphere, including the one mentioned here, are described in https://mathworld.wolfram.com/SpherePointPicking.html.

[13]This is generated by `get_ocubo.m` using a random number generator seed of 10. We expect that `five2oct.m` combined with `get_five.m` will generate near identical statistical properties to `get_ocubo.m` which is supported by a visual comparison of pairwise distance histograms (not shown in this work), and indirectly by an assertion in Section 5.3 of Morawiec [54].
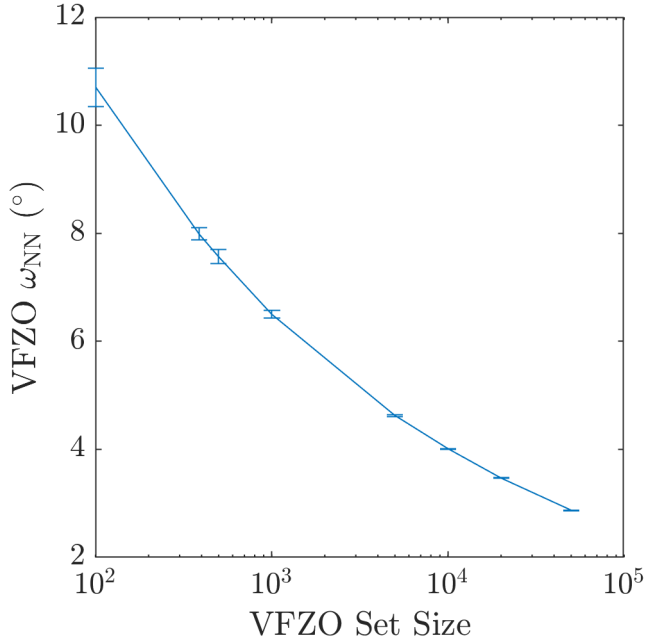
Figure 5: NN VFZ-GBO (ω_{NN}) distances (°) versus VFZ-GBO set size out of 70-80 random VFZ-GBO sets per set size.

have developed to efficiently accomplish this specialized task by pre-computing the interpolation weights (which remain fixed when only the function being evaluated changes). We then present adaptations of three other interpolation methods—GPR (Section 2.3.2), IDW (Section 2.3.3), NN (Section 2.3.4)—that are useful for general applications (an additional interpolation method—Gaussian process regression mixture (GPRM)—which we developed specifically for a non-uniformly distributed, noisy, simulation dataset is described in Section 2.4.3). Usage instructions for the VFZ-GBO repository can be found at the GitHub page (github.com/sgbaird-5dof/interp) and in Section S0.1.

2.3.1. Barycentric Interpolation

Barycentric coordinates are a type of homogeneous coordinate system that reference a prediction point within a simplex [63] or convex polytope [63–65] based on "masses" or weights at the vertices, which can be negative. The prediction point is assumed to be the barycenter (center of mass) of the simplex or convex polytope, and weights at the vertices necessary to make this assumption true are

determined. We utilize rigid SVD transformations and a standard triangulation algorithm (quickhull [55] via delaunayn() in VFZ repository function sphconvhulln.m) to define a simplicial mesh (Appendix B.1). We then use barycentric weights (i.e. coordinates) for computing intersections of a point within a simplicial facet (Appendix B.2) and for interpolation (Appendix B.3) [63]. A detailed explanation of the process is provided in Appendix B.

2.3.2. Gaussian Process Regression

GPR or Kriging uses the notion of similarity between points to fit Gaussian processes (random variables) to data based on prior information and provides uncertainty information in addition to interpolated or inferred values. For a general treatment of GPR, see Rasmussen and Williams [66]. We use MATLAB's built-in function, fitrgp(), with all default parameters[14] except that a fully independent conditional approximation is used (PredictMethod = 'fic') regardless of the number of input points. We assume a Euclidean approximation of the VFZ (see Section 2.1.3 and Figure S1). A slower, more accurate, and more memory-intensive prediction method that doesn't use sparse approximation (PredictMethod = 'exact') is also available (Section 3.2).

2.3.3. Inverse-distance Weighting Interpolation

IDW interpolation applies a weighted average to points within a neighborhood of a query point to obtain an interpolated value. interp5DOF.m implements a simple IDW approach based on [67]. A default radius of influence of $r = \sqrt{2}\mu$ is used, where μ represents the mean NN distance, and where GBO distance is approximated by the Euclidean distance or 2-norm (see Section 2.1.3, and Figure S1). NN interpolation (Section 2.3.4) is used for a given query point when there are no input points in the radius of influence.

---

[14]MATLAB R2020b was used for the Fe simulation dataset, all other results employed MATLAB R2019b, the latest installed version on our computing cluster.

2.3.4. Nearest Neighbor Interpolation

NN interpolation takes the nearest input point relative to a query point and assigns the value of the NN input point to the query point. This is implemented via the built-in MATLAB function `dsearchn()` using a Euclidean approximation of GBO distance (see Section 2.1.3, and Figure S1).

2.4. Literature Datasets

In addition to performing validation tests of the VFZ framework, we also describe results in which we apply it to actual GB property data available from literature sources. Here we briefly mention details related to the retrieval and processing of two MS simulation datasets from the literature. We describe GPR applied to Fe (Section 2.4.1) and Ni (Section 2.4.2) simulations, as well as a specialized GPRM model applied to Fe to address non-uniformity and noise concerns (Section 2.4.3).

2.4.1. Gaussian Process Regression for Fe Simulation Dataset

The Fe simulation data is obtained from [68] rather than [39] due to a mistake in the earlier dataset file[15]. GBs with a GBE less than $0.01\,\mathrm{J\,m^{-2}}$ are removed to get rid of "no-boundary" GBs. Repeated GBs are then identified and removed by converting all GBs into a VFZ-GBO set (see `Kim2oct.m`) and sorting the repeated GBs into "degenerate sets"[16], and only the average GBE (and a single GB) within each degenerate set was retained. We estimate the intrinsic RMSE and MAE of the Fe simulation dataset to be $0.065\,29\,\mathrm{J\,m^{-2}}$ and $0.061\,90\,\mathrm{J\,m^{-2}}$, respectively. Minimum and maximum error was $-0.2625\,\mathrm{J\,m^{-2}}$ and $0.2625\,\mathrm{J\,m^{-2}}$, respectively. See Section S5.2 for further details on methods used to estimate intrinsic error of the Fe simulation dataset.

---

[15]We were informed of the error during an email discussion with the corresponding author of [68].

[16]A degenerate "set" is distinct from a VFZ-GBOs "set", the former of which is discussed in greater detail in Supplementary Information Section S5.2. This sorting occurs via `avgrepeats.m` with `avgfn='min'`.

2.4.2. Gaussian Process Regression for Ni Simulation Dataset

We use the GBOs representations [53] of GBs from [42] (`'olm_octonion_list.txt'` [69]), importing and converting them to the active sense by taking the quaternion inverse of each of the GBOs' quaternions. We take GBE values (first column of `'olm_properties.txt'`, [69]), and use a GPR model (Section 2.3.2).

2.4.3. Gaussian Process Regression Mixture for Fe Simulation Dataset

Separate from the four main methods analyzed in this work, a GPRM model is developed to better predict low GBE using the non-uniformly distributed, noisy, Fe simulation dataset described in Section 2.4.1. An exponential rather than a squared exponential kernel was used for the subset GPR model (Section S5) to accommodate sharper transitions to better approximate low GBEs. Further details of the GPRM model are given in Supplementary Information (Section S5).

3. Results and Discussion

To illustrate the utility of the VFZ framework for one application, namely interpolation, we compare the (i) accuracy (Section 3.1), and (ii) efficiency (Section 3.2) of the four previously described interpolation methods implemented over the VFZ with each other and with existing methods from the literature (see Section 1). For these tests, we use the 5DOF GB energy function by Bulatov et al. [70] (trained on Ni bicrystal simulation data [42]) as a validation function which we refer to as the Bulatov Reed Kumar (BRK) function.

Following this validation study, we also demonstrate VFZ-GBO GPR interpolation applied to a large, noisy, MS Fe bicrystal simulation dataset [68] and a small, low-noise, MS Ni bicrystal simulation dataset [42] (Section 3.4.1), to evaluate performance on real GB property data.

3.1. Interpolation Accuracy

Accuracy of GPR, barycentric, NN, and IDW interpolation methods are given w.r.t. the BRK validation function (Section 3.1.1). Context is given

to these error metrics through comparison with a constant-valued control model (Section 3.1.2) and the uncertainty associated with experimental and simulated datasets (Section 3.1.3).

3.1.1. Accuracy of Four Interpolation Methods

Figure 6 provides hexagonally binned parity plots (`parityplot.m` via modified version of [71]) for each of the four interpolation methods using 50 000 input GBs. Results for 388 and 10 000 GBs are given in Figure S3 and Figure S4, respectively.

All of the methods permit successful interpolation, and the highest density region in all cases falls squarely on the parity line. The GPR and barycentric results show a slight asymmetry such that low energy values are overpredicted more often than they are underpredicted. The width of the point clouds provides a qualitative indication of the dispersion in the prediction errors, and the logarithmically scaled color indicates the frequency of errors of a given magnitude. As can be seen, the vast majority of errors are very small (the highest density—yellow region—is concentrated on the line of parity). Quantitative measures of the overall accuracy are presented for RMSE (Table 2) and MAE (Table 3), and will be discussed in detail below (see `get_errmetrics.m`).

As shown in Tables 2 and 3, of the four interpolation methods from this work, GPR has the lowest error, both in terms of RMSE and MAE, while NN has the highest error. Compared to a constant valued control model, GPR interpolation reduced the prediction RMSE by 83 %, which outperforms all of the interpolation methods in this work with respect to accuracy, as well as those considered from the literature. After GPR the next most accurate methods are barycentric, IDW, and NN We also note that the RMSE interpolation error for the GPR and barycentric methods is comparable to the minimum achievable noise-free experimental interpolation error which is the estimated error in experimental data (Section 3.1.3).

The accuracy of the predictions made using the VFZ methods depends on the VFZ-GBO set size and distribution. Figure 7 compares the prediction accuracy for each of the 4 methods to the constant valued control model, as a function of the number of input VFZ-GBOs (`ninputpts`). As expected, higher density VFZ-GBO sets result in lower error, but eventually give diminishing returns. Moreover, the standard deviations produced via multiple runs are tightly constrained and generally shrink as the VFZ-GBO set size increases.

GPR consistently gives lower error than the other three interpolation methods for all VFZ-GBO set sizes. NN interpolation produces the worst error of the four methods, but is better than a constant valued control model (i.e. average of the input GBEs) so long as `ninputpts` exceeds a few hundred input points.

It is worthwhile to note that both GPR and IDW are kernel-based in that a model parameter controls the size of the region that can influence the interpolation results. In the GPR case, this is automatically calculated via an internal fitting routine of `fitrgp()`. NN distance distributions (Figure 2) can lead to insight about correlation lengths in a given VFZ-GBO set and are used in the IDW implementation. For IDW, the radius of influence is set to $r = \sqrt{2}\mu$, where μ is the mean NN distance. It is likely that better tuning of the kernel parameters in these two methods (such as use of built-in hyperparameter optimization in the case of `fitrgp()`) could further decrease their interpolation errors. Additionally, for GPR, use of the `'exact'` `predictMethod` or a larger `'fic'` set size will also likely reduce interpolation error.

By contrast, barycentric interpolation automatically adjusts its effective region of influence because the size of the simplices in the mesh decreases as the number of vertices increases. More uniformly distributed meshes (such as obtained via constrained optimization [72, 73]) will likely result in lower, more uniform interpolation error, especially for this simplex-based approach which can exhibit high-aspect ratio facets and non-intersections outside the bounds of the mesh (Figure S7). While the barycentric interpolation error is always higher than GPR for the considered set sizes, at 50 000 VFZ-GBOs, the errors of GPR and barycentric interpolation are nearly identical.
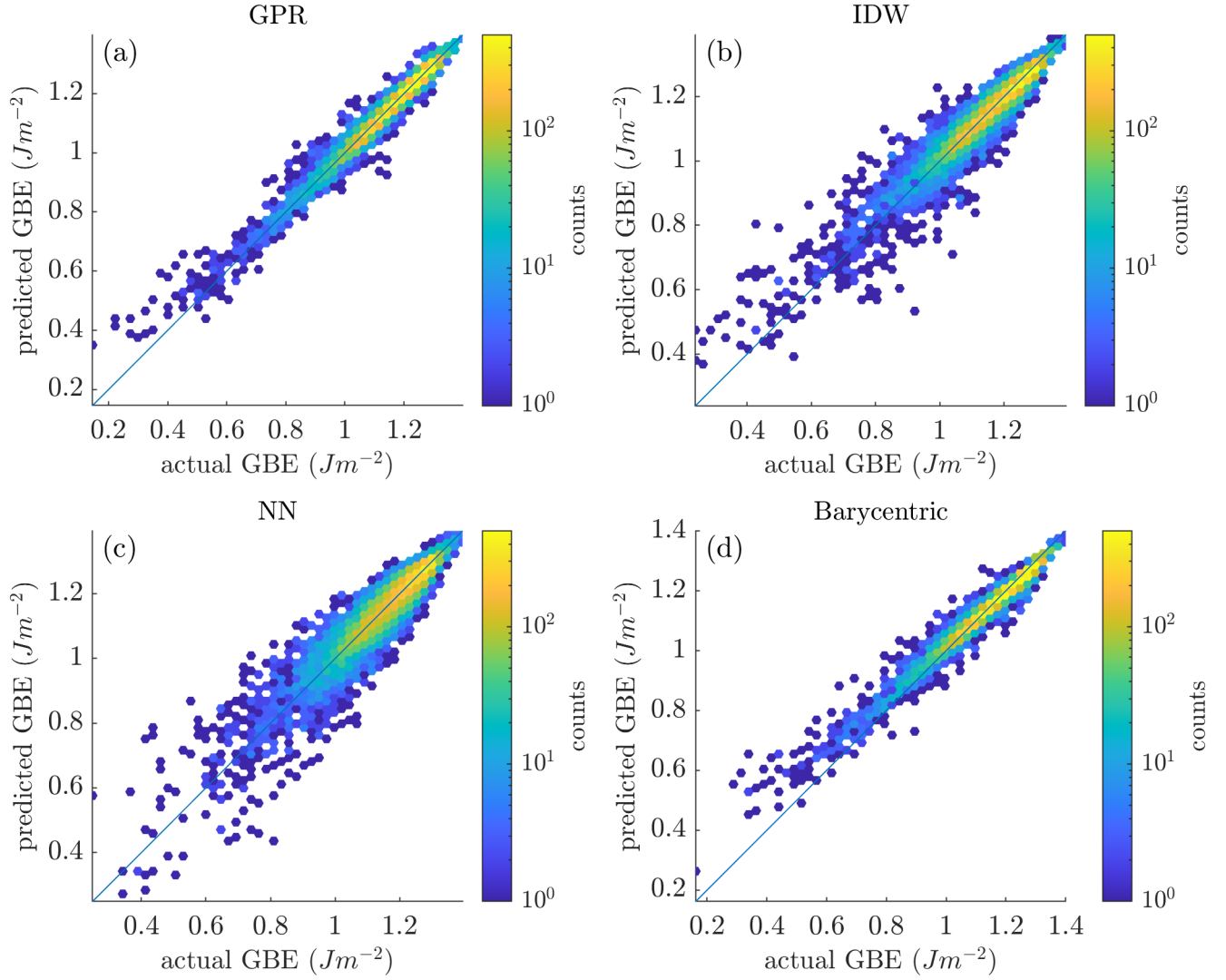
Figure 6: Hexagonally binned parity plots for 50 000 input and 10 000 prediction GBOs formed via pairs of a random cubochorically sampled quaternion and a spherically sampled random boundary plane normal. Interpolation via (a) GPR, (b) IDW, (c) NN, and (d) barycentric coordinates. BRK GBE function for face-centered cubic Ni [70] was used as the test function.

Table 2: Comparison of average interpolation RMSE (approximately 10 trial runs) for each interpolation method in the present work, using 50 000 points in the definition of the VFZ and GBEs obtained by evaluating the BRK validation function ([70]) at these points. A constant model (Cst, Avg RMSE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns represent the reduction (↓) in RMSE in absolute units of $J\,m^{-2}$ and % relative to the control model, respectively.

Method	Distance	Dataset	# GBs	RMSE $(J\,m^{-2})$	Cst, Avg RMSE $(J\,m^{-2})$	RMSE ↓ $(J\,m^{-2})$	RMSE ↓ (%)
GPR	VFZ	BRK	50 000	0.0218	0.1283	0.1065	83
Barycentric	VFZ	BRK	50 000	0.0238	0.1283	0.1045	81.4
IDW	VFZ	BRK	50 000	0.0356	0.1283	0.0927	72.3
NN	VFZ	BRK	50 000	0.0445	0.1283	0.0838	65.3

Table 3: Comparison of average interpolation MAE (approximately 10 trial runs) for each interpolation method in the present work, using 50 000 points in the definition of the VFZ and GBEs obtained by evaluating the BRK validation function ([70]) at these points. A constant model (Cst, Avg MAE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns represent the reduction (↓) in MAE in absolute units of $\mathrm{J\,m^{-2}}$ and % relative to the control model, respectively.

Method	Distance	Dataset	# GBs	MAE $(\mathrm{J\,m^{-2}})$	Cst, Avg MAE $(\mathrm{J\,m^{-2}})$	MAE ↓ $(\mathrm{J\,m^{-2}})$	MAE ↓ (%)
GPR	VFZ	BRK	50 000	0.0145	0.0955	0.081	84.8
Barycentric	VFZ	BRK	50 000	0.0145	0.0955	0.081	84.8
IDW	VFZ	BRK	50 000	0.0225	0.0955	0.073	76.4
NN	VFZ	BRK	50 000	0.0307	0.0955	0.0648	67.9
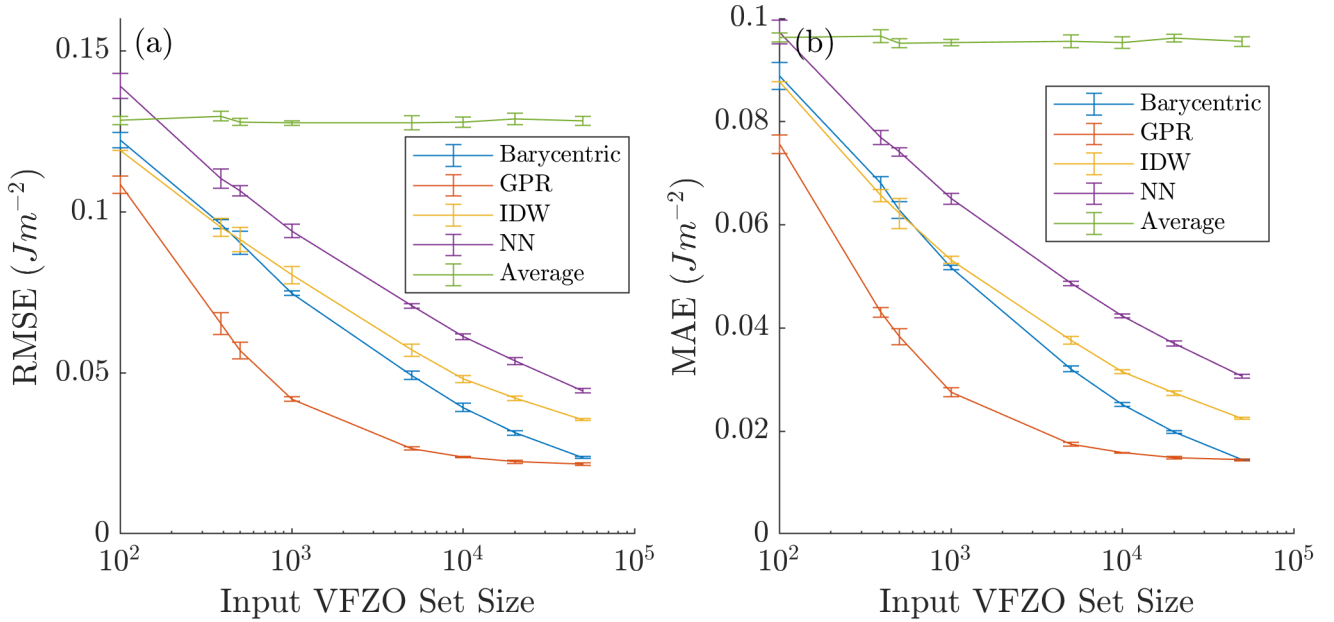
Figure 7: (a) Average RMSE and (b) average MAE vs. number of input points for (planar) barycentric (blue), GPR (orange), IDW (yellow), and NN (purple) interpolation for approximately 10 random runs with different input and prediction points. Standard deviations of approximately 10 runs are also included. Compare with approximately $0.1283\,\mathrm{J\,m^{-2}}$ and $0.0955\,\mathrm{J\,m^{-2}}$ RMSE and MAE, respectively, for a constant, average model (green) using the average of the input properties (approximately $1.16\,\mathrm{J\,m^{-2}}$).

Table 4: Approximate coordinates of VFZ-GBOs A and B used for the interpolation in Figure 8. Individual quaternions of each GBO are given in the active sense and in the laboratory reference frame with an assumed GB normal pointing in the +z direction, also in the laboratory reference frame.

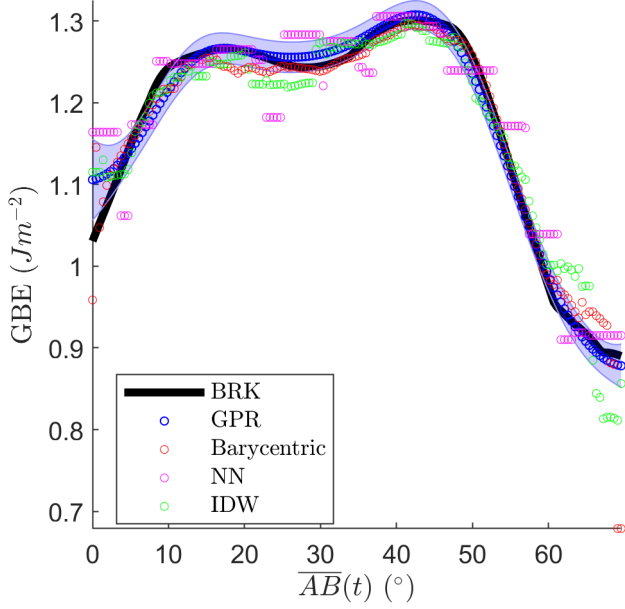| Octonion	o(1)	o(2)	o(3)	o(4)	o(5)	o(6)	o(7)	o(8)
A	0.8658	-0.4269	-0.1270	0.2280	0.2810	0.8390	-0.3852	0.2622
B	0.4684	-0.7657	-0.4100	-0.1617	-0.1483	0.8204	-0.3588	0.4198

Figure 8: Predictions of GPR (blue circles), barycentric (red circles), NN (magenta circles), and IDW (green circles) as a function of distance along a 1D arc ($\overline{AB}$) between two VFZ-GBOs (A and B). The true, underlying BRK function is also shown (black line). 50 000 random input VFZ-GBOs were generated and used for each of the models. 150 equally spaced points between A and B obtained via oSLERP [52] were used as prediction points. GPR uncertainty standard deviation is plotted as shaded error band.

3.1.2. Constant-Valued Control Models

To aid in objective interpretation of the error metrics, comparison is made to a constant valued control model, whose value is chosen to be the average of y (approximately $1.16 \, \mathrm{J\,m^{-2}}$ in the limit of ninputpts $\to \infty$) resulting in RMSE and MAE values of approximately 0.1283 and $0.0955 \, \mathrm{J\,m^{-2}}$. This comparison with the relevant constant-valued function gives a sense of the complexity and variability of the validation function and allows for a more objective comparison between differing works. For example, the RMSE for the relevant constant function compared to the validation function employed for the ANN interpolation method in [49] is $0.0854 \, \mathrm{J\,m^{-2}}$; in contrast, the RMSE for the relevant constant function compared to the BRK validation function used in this work is $0.1302 \, \mathrm{J\,m^{-2}}$ (see Table 2). This suggests that the BRK validation function is more complex and therefore less

well approximated by a constant than the validation function used to test the ANN interpolation method in [49]. Consequently, the improved performance of the present methods (see Section 3.4.1 and Tables 6 and 7) is even more notable in that the validation function employed here is more difficult to interpolate.

3.1.3. Experimental and Simulation Error

To give further context to the results of this and prior works, it is useful to consider what the intrinsic error is for typical GB property data. This provides an idea of the minimum possible interpolation error, since one cannot reliably *detect* lower error in the interpolation than already exists in the observed data itself.

One such estimate for error is furnished by the work of Shen et al. [74], who introduced a non-discretizing approach to extract relative GB energies from polycrystalline samples using the locally optimal block preconditioned conjugate gradient method. Their approach utilizes regularization imposed on triple junction equilibrium equations and k-nearest neighbor distances. Using 60 000 triple junctions (180 000 GBs) and a custom, non-smooth validation function they obtained GBE RMSE values of $0.0076 \, \mathrm{J\,m^{-2}}$ and $0.0277 \, \mathrm{J\,m^{-2}}$ for GBE values greater than $0.9 \, \mathrm{J\,m^{-2}}$ and less than $0.9 \, \mathrm{J\,m^{-2}}$, respectively. This suggests that an optimistic estimate for the error in noise-free[17] *experimental* GBE data obtained using such a method is on the order of $0.0076 \, \mathrm{J\,m^{-2}}$ to $0.0277 \, \mathrm{J\,m^{-2}}$, which also serves as an estimate of the minimum achievable noise-free experimental interpolation error for any of the interpolation methods described here. Similar analysis for noisy $0 \, \mathrm{K}$ MS *simulation* data is provided in Section 3.4 and Section S5.2 giving a RMSE and MAE of $0.065 \, 29 \, \mathrm{J\,m^{-2}}$ and $0.061 \, 90 \, \mathrm{J\,m^{-2}}$, respectively.

Again comparing the relevant constant-valued

---

[17]These errors are based on Figure 8 from Shen et al. [74], which employed synthetic triple junctions with a custom validation function, rather than experimental data. While the authors did also consider the addition of noise, we use the noise-free results as an estimate of the best-case scenario.

control model[18] to the validation function employed by Shen et al. [74], we calculate a RMSE and MAE of $0.0976\,\mathrm{J\,m^{-2}}$ and $0.0466\,\mathrm{J\,m^{-2}}$, respectively. This implies that the validation model used by Shen et al. [74] is also simpler[19] than the BRK validation model employed in the present work.

3.2. Interpolation Efficiency

Below, we present interpolation efficiency results in terms of computational runtime and memory for the four interpolation schemes used in this work (Section 3.2.1). Additionally, an in-depth treatment of the improved symmetrization runtime (separate from interpolation runtime) relative to the original GBO metric is given (Section 3.2.2).

3.2.1. Efficiency of Four Interpolation Methods

We discuss runtime and memory requirements for barycentric, GPR, IDW, and NN interpolation methods. Computational runtimes of the various methods are shown in Table 5.

Barycentric interpolation takes the longest, in spite of the fact that it is the only parallelized method by default (not accounted for in Table 5). In other words, since 12 cores were used to obtain these runtime results, the total runtime across all cores is much higher compared with the other methods; however, it is possible that other methods used multi-threading via built-in vectorized functions. The long computation times of barycentric interpolation result primarily from the large number of facets present in a high-dimensional mesh triangulation and the interconnectedness of facets with respect to each other.

GPR is fast compared to barycentric interpolation; however, the entire process has to be reevaluated (in the current implementation) if the input points (i.e. VFZ-GBOs) or input property values (i.e. GBEs) change (typically referred to as predictors/features and responses, respectively, in the

machine learning community). On the other hand, barycentric interpolation is fast if the triangulation and intersections are pre-computed and only input property values change (`interp_bary_fast.m`), but slow if the input or prediction points change, which requires recomputing the triangulation and intersections. Additionally, GPR is the second-longest in terms of of runtime.

NN and IDW interpolation have vectorized implementations and are much simpler than the barycentric and GPR methods. Consistent with expectations, NN and IDW exhibit almost negligible runtimes. It should also be noted that barycentric interpolation has much higher memory requirements than GPR, NN, and IDW due to the need to store large matrices. If `PredictMethod = 'exact'` in `fitrgp()`, then GPR also has high memory requirements for large VFZ-GBO sets. For 50 000 input points with sufficient RAM (e.g. ∼32 GB) and 12 cores available, the `'exact'` method runtime is $(535.1 \pm 392.6)\,\mathrm{seconds}$. However, because the `'fic'` approximation is always used in this work, memory requirements are similar to NN and IDW.

Because the default implementation of IDW uses a radius cut-off, the distance and weight matrices can be stored as sparse objects, dramatically reducing both the final memory storage requirements and computational complexity of this method. We expect that a k-nearest neighbor approach would produce similar results both in terms of runtime and error when a relatively uniform sampling of GBC is obtained.

3.2.2. Symmetrization Runtime Comparison with Traditional Octonion Metric

In addition to the interpolation runtime of the methods just presented, it is valuable to consider the runtime of the VFZ symmetrization step (not included in Table 5). The symmetrization step is at the core of the VFZ framework and is a key to its overall performance. It is a common step for both (i) distance calculations and (ii) all of the interpolation methods presented here.

Directly computed, scaled Euclidean and arc length distances in the VFZ framework approximate the original GBO distance by Francis et al. [52], and the calculation speed is even higher than

---

[18]We use the mean of the true GBEs from their validation function to define the constant-valued control model instead of the mean of the input GBEs because the latter does not exist for polycrystalline data.

[19]Shen et al. [74] used 8 cusps of varying depths and widths based on the Read-Shockley model and unity GBE everywhere else.

Table 5: Comparison of average runtime (s) for 10 trials for barycentric, GPR, IDW, and NN interpolation methods for various input VFZ-GBO set sizes using 12 cores and evaluated on 10 000 prediction VFZ-GBOs. Because GPR, IDW, and NN method defaults do not use `parfor` loops but may have internal multi-core vectorization, it is unclear to what extent the number of cores affects the runtime of methods other than barycentric interpolation. VFZ-GBO symmetrization runtime was not included; however, symmetrization of 50 000 GBOs takes approximately 76 seconds on 6 cores (Intel i7-10750H, 2.6 GHz) and is a common step in every interpolation method (i.e. it is fundamental to the VFZ framework). We used the BRK validation function for GBE [70].

VFZ-GBO Set Size	Runtime (s)			
	Barycentric	GPR	IDW	NN
100	191.8 ± 19.57	0.4187 ± 0.4342	0.034 ± 0	0.0367 ± 0.0041
388	388.4 ± 18.84	0.943 ± 0.3481	0.0904 ± 0.0224	0.0705 ± 0.0129
500	455.7 ± 55.28	0.6104 ± 0.3138	0.1352 ± 0.0364	0.0724 ± 0.0051
1000	536.5 ± 35.26	1.743 ± 0.9464	0.1948 ± 0.0395	0.1203 ± 0.0184
5000	998.9 ± 54.48	5.216 ± 0.4816	0.8726 ± 0.1529	0.9277 ± 0.2418
10 000	1516 ± 56.59	5.609 ± 0.8756	1.631 ± 0.3915	0.8938 ± 0.1717
20 000	2526 ± 119.5	11.45 ± 3.29	3.191 ± 0.4752	1.275 ± 0.3423
50 000	5743 ± 361.3	13.69 ± 4.05	7.635 ± 1.872	3.817 ± 0.5884

explicit GBO distance calculations using the original GBO distance. For example, 50 000 GBOs can by symmetrized into VFZ-GBOs in approximately 76 seconds using 6 cores (`get_octpairs.m`), and the corresponding $50\,000 \times 50\,000$ pairwise-distance matrix can be computed in approximately 10 seconds (`pdist()`), giving a total runtime of approximately 86 seconds (466 total CPU seconds). Compared to the original GBO metric distance calculations [53] in the Fortran-based EMSoft package [75], this represents an improvement in computational speed by ~5 orders of magnitude using our MATLAB implementation in the VFZ repository [60].

Improvement per distance calculation per core of the VFZ repository is about 4×10^5 relative to the EMSoft [75] metric of 26 minutes using 8 cores for a 388×388 pairwise distance matrix. This EMSoft timing information is directly reported in Chesser et al. [53]. In other words, computation of a $50\,000 \times 50\,000$ using the traditional GBO metric and EMSoft implementation would take approximately 6.6 CPU years (or 153 CPU days by applying the isometry equation in Section 7 of Morawiec [54]). Since most interpolation methods will depend on computing new distances, probing the model at new GBs will also be expensive. For example, it would take at minimum ~30 CPU days (after isometry equivalence has been applied) to

perform property interpolation for 10 000 prediction GBs assuming the pairwise-distance matrix relative to 50 000 input GBOs needs to be computed. This presents an issue for iterative simulations (e.g. mesoscale grain growth) in which 1000's of new GB segments would need to be sampled at each time step. By contrast, property values for 10 000 new GBs would be sampled in our approach in ~90 CPU seconds. For perspective, a phase-field simulation might have 10 000 or more time steps with thousands of GBs Kim et al. [68], Dimokrati et al. [76]. Recently, Miyoshi et al. [77] presented Reed-Shockley anisotropic 3D phase-field grain growth results for initially 3 125 000 grains with as many as 125 000 time steps to reach ~10 000 final grains. Performing such a simulation with even the efficient VFZ framework would require 56 CPU years for the property sampling alone [20].

This significant speed up stems from the fact that in the VFZ framework SEOs only need to be considered once per GB, $O(L)$, rather than once per distance calculation, $O(L^2)$, and that SEOs only

---

[20]For such an application, a GPU implementation of the VFZ framework, batch implementation of the SEO considerations, directly tracking GBs movement within a VFZ, and/or other approaches would likely be necessary to make the problem more tractable.

need to be considered once in a GB pair, $O(N_p^2)$, rather than for every combination between the two GBs, $O(N_p^4)$. The SEO computation complexity is thus $O(N_p^2 L)$, a significant improvement compared with the original SEO complexity of $O(N_p^4 L^2)$ [53], where N_p is the number of proper rotations of the crystallographic point group ($N_p = 24$ for $m\bar{3}m$ face-centered cubic point group) and L is the number of GBs.

Empirically, to compute a pairwise-distance matrix for $L = 50\,000$ GBs using the VFZ repository [60], the full $O(N_p^2 L)$ symmetrization operations take about $76\,\text{seconds} \times 6\,\text{cores} = 456\,\text{seconds}$ of CPU time, whereas the subsequent pairwise-distance computation is $O_{\text{pd}}(L^2)$ and takes approximately $10\,\text{seconds}$ for a $50\,000 \times 50\,000$ matrix. Even though $O(N_p^2 L) \ll O_{\text{pd}}(L^2)$, the symmetrization step takes far more time than the pairwise distance calculation (even for large L) because of the cost of generating SEOs. Because Euclidean distances—which can be computed faster than trigonometric inverse functions—are employed, and built-in, vectorized MATLAB functions are utilized, there is a further speed enhancement in the VFZ-GBO approach.

3.3. Interpolation Visualization

We present interpolation results plotted in a 1D arc in the full 5DOF GB space (Section 3.3.1) followed by discussion of potential to use numerical derivatives and identify local minima (Section 3.3.2).

3.3.1. Interpolation Along a 1D Arc

To provide a visual illustration of the property predictions, Figure 8 shows the predicted GBE for each of the four interpolation methods as a function of distance along a 1D arc ($\overline{AB}$) between two VFZ-GBOs, A and B. Approximate coordinates for A and B are given in Table 4, and each intermediate point between A and B resides on the surface of a hypersphere. The 150 intermediate points were obtained using oSLERP [52]. Each model used its own set of $50\,000$ random input VFZ-GBOs with GBE sampled via the BRK validation function. The two VFZ-GBOs were chosen by taking the furthest apart pair out of $20\,000$ VFZ-GBOs

which thus approximates the largest dimension of the VFZ where each endpoint is close to the true VFZ exterior.

Comparison of the predictions from the four interpolation methods with the true values of the BRK validation function along this 1D path shows that all methods yield reasonable agreement with the true model. The GPR and barycentric methods appear to agree most with the true model, followed by IDW and NN. The NN method shows the piecewise-constant (stair-step) artifact typical of NN methods. We also note that while the fidelity of the predictions is quite good for all methods in the interior of the VFZ, the performance does degrade at the extreme limits of the VFZ (note the deviations at the left and right limits of Figure 8). This effect seems to be particularly pronounced for the barycentric method, and much less so for the GPR method.

We believe this is the first[21] plot of a GB property continuously interpolated between two arbitrary GBs (i.e. neither residing entirely in a single misorientation fundamental zone nor a single boundary plane fundamental zone). Such visualizations can naturally be extended to 2D and 3D by plotting colored points in a triangle or tetrahedron, respectively, all of which (1D, 2D, and 3D) represent small "slices" of the GBC space.

3.3.2. Potential for Numerical Derivatives

Additionally, such visualizations suggest the ability to estimate numerical derivatives or gradients of GB properties without being restricted to a GB subspace (e.g. misorientation fundamental zone or boundary plane fundamental zone) which can be a useful mathematical construct for the GB community. For example, steepest descent paths and all local GBE minima can be estimated and used in grain growth simulations. In such contexts, use of en-

---

[21]OSLERP results from Francis et al. [52] plots GB *structure* continuously between two GBs, Chesser et al. [53] performs cross-validation on the simulated Olmsted Ni GBs, and [54] plots distances between GBs on a geodesic with another GB. The results in these works are distinct from what is presented here: a plot of continuously interpolated GBEs between two arbitrary GBs.

sembled VFZ-GBO interpolation may be necessary to mitigate discontinuity artifacts when crossing the exterior of a VFZ as discussed in Section 2.1.3 which we plan to explore in future work.

3.4. Literature Datasets

In addition to validation results (Section 3.1), we also apply the VFZ framework to real GB property data from two sources in the literature. This allows more direct comparison to previous methods as well as demonstration of the performance of the the VFZ framework for typical MS data. Specifically, we present GPR interpolation results for MS Fe and Ni simulation datasets and compare them with prior work (Section 3.4.1). Finally, because GPR overestimates the low GBE for the non-uniformly distributed, noisy Fe simulation dataset, we also provide results for an adaptation called the GPRM model that compensates for this effect (Section 3.4.2).

3.4.1. Comparison with Prior Work

The GPR interpolation method of the present work was used with the same number of input GBs as was supplied in Restrepo et al. [49] for Fe (17 176) and Chesser et al. [53] for Ni (388) to provide a more consistent comparison with prior work. For Fe, the remainder of the simulation data was used for testing, consistent with Restrepo et al. [49], except that zero-energy GBs and degenerate GBs were treated differently as described in Section 2.4.1. For Ni, a leave-one-out cross validation scheme was used, consistent with Chesser et al. [53].

Hexagonally binned parity plots for the Fe and Ni simulation datasets are shown in Figure S8d and Figure S12, respectively. RMSE and MAE comparisons along with improvement relative to a constant, average model are given in Table 6 and Table 7, respectively.

For the Fe case, we see a larger improvement than prior work likely due to our incorporation of GB symmetry, which was not considered in Restrepo et al. [49]. For the Ni case, there is a slight improvement relative to prior work, indicating that accuracy is similar to the original GBO metric while maintaining the significant computational benefits of the VFZ framework.

Since the BRK validation function is also an interpolation function on the Ni simulation data, GPR within the VFZ framework and the BRK function results are directly compared via parity plot in Figure 9.

For the BRK and GPR interpolations, MAE is $0.009\,75\,\mathrm{J\,m^{-2}}$ and $0.036\,26\,\mathrm{J\,m^{-2}}$, respectively. Likewise, RMSE is $0.017\,27\,\mathrm{J\,m^{-2}}$ and $0.049\,72\,\mathrm{J\,m^{-2}}$, respectively. From Figure 9a, we see that low GBE is predicted more accurately and high GBE less accurately with BRK interpolation vs. GPR in the VFZ framework. Without access to the original fitting routines used to produce the BRK function, we have not performed leave-one-out cross validation which would allow for a safer model evaluation (i.e. one in which fair results are less likely due to overfitting). Leave-one-out cross validation results for the GPR case are, however, shown in Figure S12, indicating that the model performs much worse in such a data-limited regime at points the model has never seen before.

3.4.2. Gaussian Process Regression Mixture Applied to Metastable Fe Simulation Data

In addition to GPR, a GPRM model (Figure S8) based on a sigmoid mixing function (Figure S9) is used to better predict low GBE values of the non-uniformly distributed, noisy Fe dataset (Section 2.4.3)[22]. GPRM interpolation results for the Fe GBE simulations [68] are shown in Figure 10, where approximate coordinates for the GBOs A and B in Figure 10b are given in Table 8.

We find that:

- the model error is on par with the intrinsic error of the data

- the predictions likely exhibit overprediction bias relative to the true minimum for a given GB

- future availability of multiple metastable state GBEs is anticipated to greatly improve the model performance

---

[22]Alternatively, including no-boundary octonions may likewise improve low GBE performance, but possibly at the expense of high GBE predictive accuracy.

Table 6: Comparison of interpolation MAE (1 trial run) for $0\,\mathrm{K}$ molecular statics (MS) datasets. A constant model (Cst, Avg MAE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns, MAE $\downarrow$ ($\mathrm{J\,m^{-2}}$) and MAE $\downarrow$ (%)), represent the reduction in MAE in units of $\mathrm{J\,m^{-2}}$ and % relative to the control model, respectively. Non-sym refers to distances calculated in Restrepo et al. [49] without regard for crystal symmetries.

Method	Distance	Dataset	# GBs	MAE ($\mathrm{J\,m^{-2}}$)	Cst, Avg MAE ($\mathrm{J\,m^{-2}}$)	MAE $\downarrow$ ($\mathrm{J\,m^{-2}}$)	MAE $\downarrow$ (%)
GPR	VFZ	MS Fe	17 176	0.0405	0.0617	0.0212	34.4
ANN [49]	Non-sym	MS Fe	17 176	0.0486	0.0617	0.0131	21.2
Laplacian kernel regression [53]	GBO	MS Ni	388	—	0.1752	—	—

Table 7: Comparison of interpolation RMSE (1 trial run) for $0\,\mathrm{K}$ molecular statics (MS) datasets. A constant model (Cst, Avg RMSE), whose value was chosen to be the mean of the input GBE was used as a control. The last two columns, RMSE $\downarrow$ ($\mathrm{J\,m^{-2}}$) and RMSE $\downarrow$ (%)), represent the reduction in RMSE in units of $\mathrm{J\,m^{-2}}$ and % relative to the control model, respectively. Non-sym refers to distances calculated in Restrepo et al. [49] without regard for crystal symmetries.

Method	Distance	Dataset	# GBs	RMSE ($\mathrm{J\,m^{-2}}$)	Cst, Avg RMSE ($\mathrm{J\,m^{-2}}$)	RMSE $\downarrow$ ($\mathrm{J\,m^{-2}}$)	RMSE $\downarrow$ (%)
ANN [49]	Non-sym	MS Fe	17 176	—	0.0854	—	—
GPR	VFZ	MS Ni	388	0.0951	0.2243	0.1292	57.6
Laplacian kernel regression [53]	GBO	MS Ni	388	0.0977	0.2243	0.1266	56.4
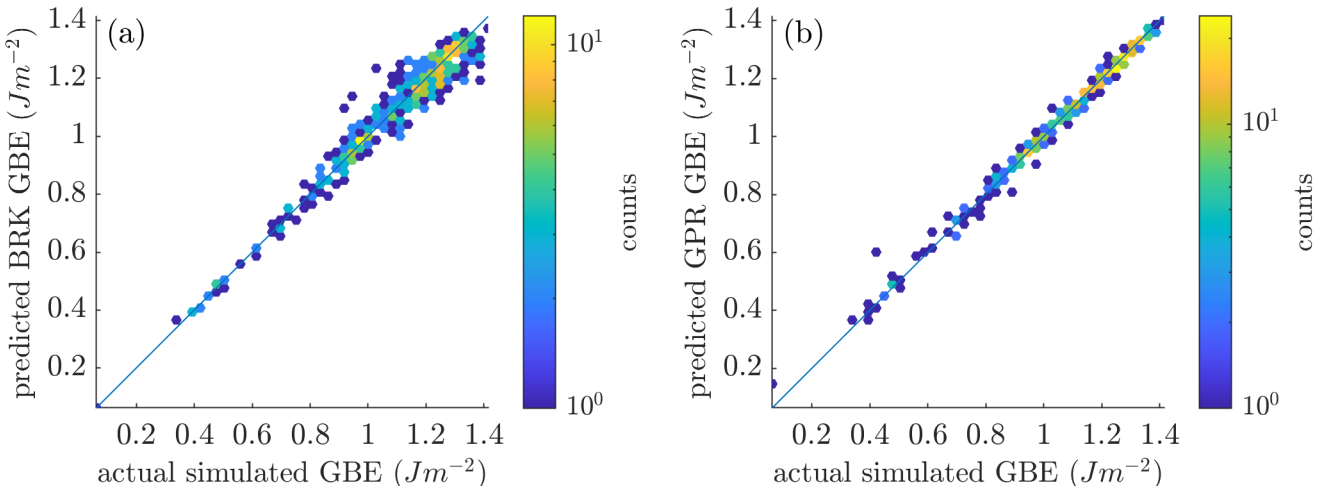
Figure 9: Hexagonally binned parity plots of (a) BRK and (b) GPR model GBEs fitted using Olmsted Ni simulation data vs. Olmsted Ni simulation GBEs. MAE is $0.009\,75\,\mathrm{J\,m^{-2}}$ and $0.036\,26\,\mathrm{J\,m^{-2}}$ for (a) and (b), respectively. Likewise, RMSE is $0.017\,27\,\mathrm{J\,m^{-2}}$ and $0.049\,72\,\mathrm{J\,m^{-2}}$, respectively.

We now elaborate each of these points.

First, because only a single metastable state was used for each GBE simulation, both the training and validation data are subject to noise, consistent with a wide lateral spread of predictions in both Figure 10 and the intrinsic error estimation (Figure S11). The Fe simulation dataset GPRM model gives lower RMSE ($0.055\,035\,\mathrm{J\,m}^{-2}$) and MAE ($0.039\,185\,\mathrm{J\,m}^{-2}$) than the intrinsic error estimates. This indicates that the intrinsic error itself is somewhat overestimated[23]. The fact that both model and intrinsic error metrics are relatively close and the prediction and intrinsic error parity plots (Figure 10 and Figure S11b, respectively) are similar suggests that the model is performing well. It also suggests that further improvements in the model relative to the "true" values will be "hidden", i.e. they will probably not manifest as lower RMSE or MAE nor as more tightly distributed parity plots, etc.

Next, given the theoretical existence of a true minimum GBE for a given GB, the predictions which were based on metastable GBEs can be assumed to have an overprediction bias relative to the true minimum. On average, we expect this overprediction bias relative to the true minimum GBE (rather than the most likely metastable state) may be on the order of a few hundred $\mathrm{mJ\,m}^{-2}$ and may vary as a function of true minimum GBE. In other words, the model obtained is probably an estimate of the most likely metastable GBE rather than the true minimum GBE. This is akin to saying that we obtain from this data a model that approximates the non-equilibrium, Stillinger quenched red curve of Figure 4(c1) in [27], not the minimum GBE blue curve of the same chart. See [27] for an in-depth treatment of equilibrium and metastable GBE.

Finally, datasets where multiple metastable GBEs (e.g. 3-10 repeats) are provided for each GB will likely greatly improve the performance of the GPR model in predicting either the most likely metastable GBE (when all GBEs are considered) or the true minimum GBE (when only the minimum GBE is considered for each GB) and may even negate the need for a GPRM approach. Thus, it is suggested that, where feasible, future large-scale GB bicrystal simulation studies will report all property data for repeated trial runs rather than a single trial run or a single value from a set of trial runs. Ideally, data for the three additional microscopic DOFs for GBs (which falls into the category of epistemic uncertainty in this work) would also be included. We believe it is likely that minimum energy paths (i.e. paths of steepest descent) in the GBE landscape depend on both macroscopic and microscopic DOFs (in total, 8DOF) and could offer a more holistic view of GB behavior that better mimics and explains experimental grain growth observations. Indeed, it has been experimentally observed that at least some GB migration mechanisms involve structural transformations between equilibrium GBs via metastable states [28].

4. Conclusion

In this work, we presented the VFZ framework for (i) computing distances between GBs and (ii) predicting the properties of GBs from existing measurements. We found that distance calculations in the VFZ framework are dramatically more computationally efficient than traditional methods at the expense of infrequent, large distance overestimation which can be addressed through ensemble techniques at a small computational cost as discussed in Section 2.1.3.

We also developed and tested a barycentric interpolation method, and adapted three other interpolation methods for use in the VFZ framework. We provide an easy-to-use, versatile implementation of our methods through an interpolation function `interp5DOF.m` written in MATLAB (github.com/sgbaird-5dof/interp, [60]) and many companion functions in the VFZ repository. This approach is general and can be applied to any crystal system (any of the 32 crystallographic point

---

[23]The prediction error of a model typically cannot be less than the noise of the prediction data of a model even if the model is estimating the true prediction values with better accuracy than the noise (which is very possible and even expected with GPR models when the noise in the input data is approximately Gaussian).
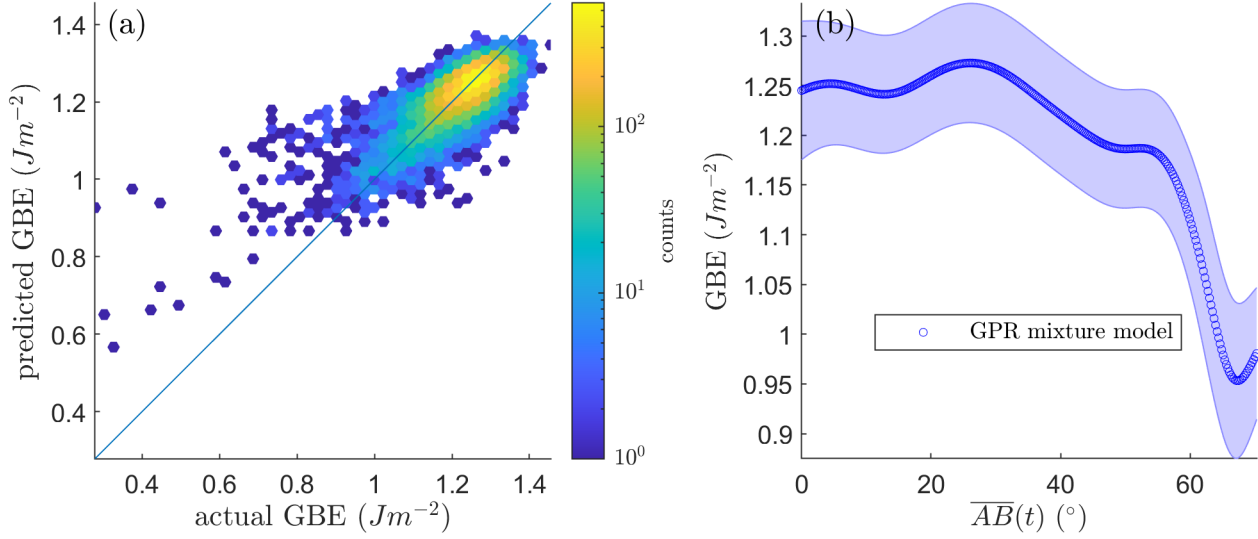
Figure 10: Interpolation results for a large Fe simulation database [68] using 46 883 input GBs and 11 721 prediction GBs in an 80%/20% split and a GPRM model to better approximate low GBEs. Use of a GPRM model predicts low GBE better than the standard GPR model (compare with Figure S8d). (a) Hexagonally binned parity plot of the GPR mixing model with RMSE and MAE of $0.055\,035\,\mathrm{J\,m^{-2}}$ and $0.039\,185\,\mathrm{J\,m^{-2}}$, respectively, relative to typical, constant average models of $0.0854\,\mathrm{J\,m^{-2}}$ and $0.0617\,\mathrm{J\,m^{-2}}$, respectively. (b) Predictions of GPRM model (blue circles) as a function of distance along a 1D arc $(\overline{AB})$ between two VFZ-GBOs (A and B).

Table 8: Approximate coordinates of VFZ-GBOs A and B used for the MS Fe simulation dataset interpolation in Figure 10. Individual quaternions of each GBO are given in the laboratory reference frame with an assumed GB normal pointing in the +z direction, also in the laboratory reference frame.

Octonion	o(1)	o(2)	o(3)	o(4)	o(5)	o(6)	o(7)	o(8)
A	0.8716	-0.4124	-0.1857	0.1893	0.3146	0.8359	-0.3815	0.2382
B	0.4391	-0.7856	-0.4142	-0.1360	-0.1376	0.8082	-0.3705	0.4366

22

groups can be selected by the parameter `pgnum`[24]). The methods described here may be applicable to other distance metrics (see Morawiec [54] for a comprehensive summary of metrics). We also developed a GPRM model specifically for better low GBE prediction using a non-uniformly distributed, noisy dataset.

Of the interpolation methods that we present in this work, GPR provided the highest accuracy predictions. It also provided higher accuracy predictions than any of the methods in the literature. The GPR interpolation errors (50 000 VFZ-GBOs) for the BRK validation model are about 2.4 times the intrinsic error that would be expected from reconstruction of noise-free, experimental polycrystalline data via locally optimal block preconditioned conjugate gradient [74] (180 000 GBs) with their simpler validation model. Moreover, the interpolation errors for a Fe simulation dataset are on par with the intrinsic errors of the dataset itself (Section S5.2). While IDW and NN interpolation have the fastest computation times, they also have higher interpolation error. Consequently, we recommend the GPR interpolation method for the VFZ framework for most applications because it provides the best combination of accuracy and speed and handles input noise; however, the other methods can meet niche needs. For example, barycentric interpolation enables rapid and accurate predictions when the function to be evaluated changes, but the input and prediction GBs remain fixed.

We anticipate that the VFZ framework and corresponding implementation will benefit numerous applications related to GB structure and properties, including facilitating GB structure-property model development, enabling efficient surrogate modeling of GB properties, and larger scale iterative simulations that require repetitive evaluation of computationally expensive structure-property models.

Acknowledgement

CRediT Statement

Sterling Baird: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Oliver Johnson**: Supervision, Project administration, Funding acquisition, Conceptualization, Writing - Review & Editing. **David Fullwood**: Funding acquisition, Writing - Review & Editing. **Eric Homer**: Funding acquisition, Writing - Review & Editing

Appendix A Active vs. Passive Convention

Misorientation quaternions are represented in the active sense[25]:

$$q_m = q_A{}^{-1} q_B \qquad (4)$$

where q_m, q_A, and q_B represent the misorientation quaternion, orientation quaternion of grain A in the sample frame, and orientation quaternion of grain B in the sample frame, respectively. The $^{-1}$ operator denotes a unit quaternion inverse (identical

---

[24]While our testing focused on cubic point group symmetry, symmetry operators for other point group symmetries were provided in the TutorialCode/crystal_symmetry_ops directory of `github.com/ichesser/GB_octonion_code` (as of commit: f57f9be). Other point groups (in particular those which are noncentrosymmetric) may give rise to differently shaped/larger VFZs for which a Euclidean distance approximation will have the worst case error of 2 vs. the true value of π which represent the furthest Euclidean and arc length distances on a unit hypersphere, respectively. The distance type of `GBdist4.m` can be changed from `'norm'` to `'omega'` to address this issue. We plan to investigate symmetries other than cubic in future work.

---

[25]The passive convention is used in [52]

to conjugation of a unit quaternion). Quaternion multiplication is given by equation 23 of [78]

$$pq \equiv (p_0 q_0 - \mathbf{p} \cdot \mathbf{q}, q_0 \mathbf{p} + p_0 \mathbf{q} + P\mathbf{p} \times \mathbf{q}) \qquad (5)$$

where q_0 and p_0 are scalar components of the quaternions, and $\mathbf{q}$ and $\mathbf{p}$ are the vector components.

In this work, we use the convention that $P = 1$ throughout the various operations in the VFZ repository ($P \equiv$ epsijk) and highly encourage interested readers to refer to Rowenhorst et al. [78] to understand the redefined versions of quaternion multiplication, quaternion rotation, nuances associated with use of active vs. passive conventions, etc. Boundary plane unit normals are expressed pointing away from grain A and in the reference frame of grain A (i.e. the outward-pointing normal convention).

Appendix B Detailed Barycentric Interpolation Method

We describe barycentric interpolation applied in the VFZ framework in more detail. This includes:

1. triangulation of a VFZ mesh (Appendix B.1)

2. finding intersections between arbitrary VFZ-GBOs and the VFZ mesh (i.e. finding intersecting facets) (Appendix B.2)

3. calculating interpolated values of an arbitrary VFZ-GBO property using the intersecting facet (Appendix B.3)

B.1 Triangulating a Voronoi Fundamental Zone Mesh

Creation of a simplicial mesh is necessary to perform barycentric interpolation. Due to the difficulty of visualizing a 7-sphere, we provide visual illustrations of the process as applied to lower-dimensional analogues. After GBOs have been symmetrized into a VFZ (Section 2.1.1), the triangulation process occurs by:

1.1 applying a SVD transformation to remove the U(1)-symmetry degeneracy inherent in the VFZ-GBO coordinates (Appendix B.1.1)

1.2 linearly projecting VFZ-GBOs onto a hyperplane that is tangent to the vector between the origin and the mean of the input VFZ-GBOs to reduce computational burden of the triangulation

1.3 performing a second SVD transformation (Appendix B.1.3)

1.4 computing the triangulation according to the quickhull algorithm [55] using built-in methods

In the explanation of each of these steps that follows, we make reference to lower-dimensional visual analogues of the VFZ-GBO triangulation procedure, which are given in Figure B.1, Figure B.2, and Figure B.3. We note that 3D Cartesian coordinates in Figure B.1 correspond to 8D Cartesian coordinates, whereas 3D Cartesian coordinates in Figure B.2 and Figure B.3 correspond to 7D Cartesian coordinates. This is intentional for two reasons:

- Figure B.1 illustrates that unsymmetrized 8D Cartesian GBOs are analogous to a point cloud on the 2-sphere (Figure B.1a) and that an 8D Cartesian VFZ-GBO set, which has already been symmetrized, is analogous to a geodesic arc on the 2-sphere (Figure B.1b). A VFZ-GBO set has a degenerate dimension that can then be removed by a rigid SVD transformation to 7D Cartesian coordinates (analogous to 2D Cartesian coordinates in Figure B.1c). This sequence would be more difficult to visualize if Figure B.1a was meant to represent a point cloud on the 3-sphere (4D Cartesian coordinates), etc.

- Figure B.2 illustrates a second transformation from normalized 7D Cartesian coordinates (Figure B.2a) to a hyperplane (Figure B.2b) which is then transformed into 6D Cartesian coordinates via a second SVD. In this case, key issues are retained that would otherwise be lost (Section S4.1) if an arc on a circle (1-sphere) to 1D Cartesian coordinates were used

instead[26]. Additionally, the use of actual triangles is a more familiar and compelling illustration of *triangulation.*

While lower dimensional analogues are useful for visualizing and understanding the process of triangulation, a written description is also given in the following sections. As appropriate, we refer back to the teaching figures described in this section.

B.1.1 Singular Value Decomposition Transformation from 8D Cartesian to 7D Cartesian

To reduce the computational complexity of triangulating a high-dimensional mesh [55], some simplifications are made. First, the degenerate GBO dimension obtained from analytically minimizing $U(1)$ symmetry [52] is removed via a rigid (i.e. distance- and angle-preserving) SVD transformation, analogous to a Cartesian rotation and translation (see 3D to 2D SVD transformation from Figure B.1b to Figure B.1c).

B.1.2 Linearly Project onto Hyperplane

Next, the resulting 7D Cartesian representation of each VFZ-GBO is projected onto a hyperplane that is tangent to the centroid (i.e. mean) of the VFZ-GBO set[27] (Figure B.2a). By performing this linear projection, one of the dimensions becomes degenerate.

B.1.3 Singular Value Decomposition Transformation from 7D Cartesian to 6D Cartesian

This additional degeneracy is removed via a second SVD transformation, this time to 6D Cartesian coordinates (see 3D to 2D projection in Figure B.2a-b). Finally, the resulting points can be triangulated via the quickhull algorithm [55] (see VFZ repository function `sphconvhulln.m` and built-in

MATLAB function `delaunayn()`), which relies on Euclidean distances[28]. Because the simplicial mesh is defined by a list of edges between vertices for each simplicial facet, this list applies immediately to the VFZ-GBO set in its 7D Cartesian coordinates (i.e. no reverse transformation is necessary to use the mesh on the 6-sphere in 7D).

B.2 Intersections in a Voronoi Fundamental Zone Mesh

Once the triangulation has been determined, we need to find which facet each prediction point intersects (i.e. find the intersecting facet). There are two sub-steps:

2.1 applying the same rigid transformation to the prediction points as was applied to the input points (otherwise the prediction points won't line up properly with the mesh) (Appendix B.2.1)

2.2 identifying facets nearby a prediction point and testing for intersection (Appendix B.2.2).

B.2.1 Apply Same Singular Value Decomposition to Input and Prediction Points

The positions of the prediction points need to be fixed relative to the mesh even after the rigid SVD transformation. This is accomplished by:

2.1a concatenating both input and prediction points

2.1b using the `interp5DOF.m` sub-routine `proj_down.m` (which depends on MATLAB's built-in SVD implementation `svd()`) to perform the transformation

2.1c subsequently separating the transformed input and prediction points (reverse of concatenation step)

---

[26]Non-intersection issues due to high-aspect ratios and consideration of facets connected up to `nnMax` NNs do not manifest in triangulations on the surface of a 1-sphere because one of the two facets (i.e. line segments) connected to the first NN mesh vertex relative to the prediction point is guaranteed to have an intersection.

[27]This is *not* a rigid transformation; however, it approximates one with sufficient accuracy to produce a high-quality triangulation in a VFZ.

[28]While the triangulation algorithm used in this work relies on Euclidean distances (the use of which is possible via the VFZ framework), other distance metrics that are non-Euclidean [54] could potentially be incorporated into the barycentric approach such as by doing an edge-length based simplex reconstruction [79, 80] using the VFZ triangulation edge lengths.
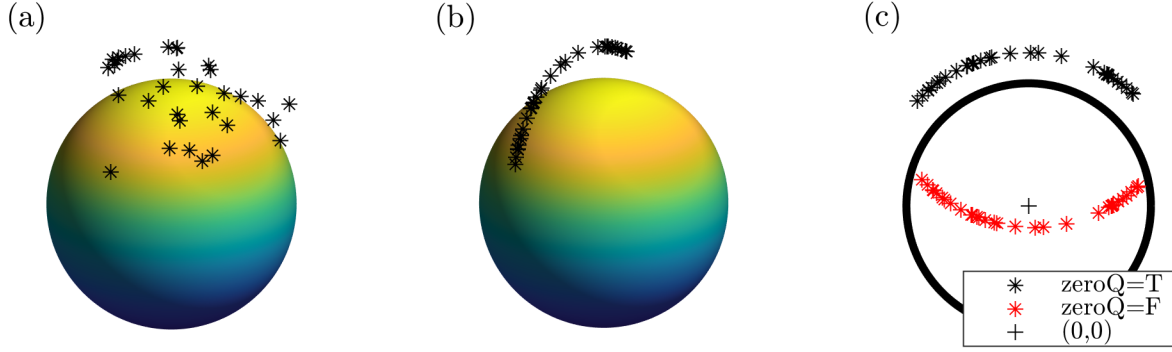
Figure B.1: 3D Cartesian to 2D Cartesian analogue of 8D Cartesian to 7D Cartesian degeneracy removal via rigid SVD transformation as used in barycentric interpolation approach. (a) Starting spherical arc points on surface of 2-sphere, (b) rotational symmetrization applied w.r.t. z-axis (analogous to U(1) symmetrization), and (c) degenerate dimension removed via singular value decomposition transformation to 2D Cartesian with either the origin (black plus) preserved (black asterisks, `zeroQ=T`) for triangulation or ignored (red asterisks, `zeroQ=F`) for mesh intersection. The spheres (a,b) and circle (c) each have a radius of 0.8 and are used as a visualization aid only.
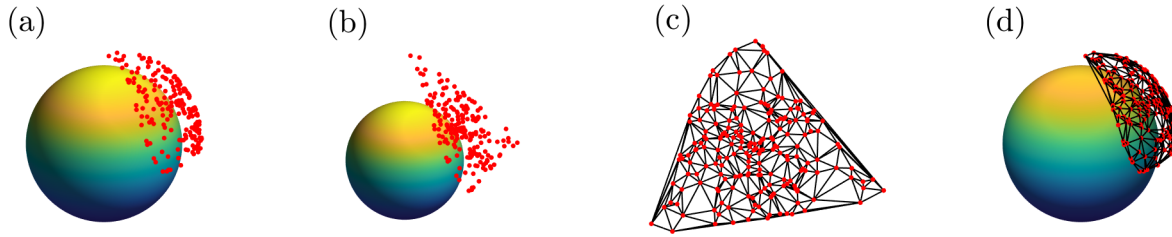
Figure B.2: 3D Cartesian to 2D Cartesian analogue of 7D Cartesian to 6D Cartesian mesh triangulation used in barycentric interpolation approach. (a) 3D Cartesian input points are (b) linearly projected onto hyperplane that is tangent to mean of starting points. (c) The degenerate dimension is removed via a rigid SVD transformation to 2D Cartesian and the Delaunay triangulation (black lines) is calculated, with input vertices (red). Delaunay triangulation superimposed onto normalized input points (d). The spheres in (a), (b), and (d) have a radius of 0.8 and are used for visualization only.

To map new points onto the mesh, the `usv` structure output from `proj_down.m` needs to be stored and supplied in future calls to `proj_down.m`. Likewise, `usv` need to be supplied to `proj_up.m` to perform the reverse SVD transformation.

B.2.2 Testing Nearby Facets for Intersections

Once the prediction points are lined up properly with the mesh, the facet containing the prediction point (i.e. intersecting facet) is found. We define the intersecting facet as the one for which a point's barycentric coordinates are positive within a given tolerance. Consequently, we determine facet affiliation by:

2.2a linearly projecting the prediction point onto the hyperplane defined by a mesh facet's vertices (Figure B.3)

2.2b computing the point's barycentric coordinates within the facet [81, 82] (see VFZ repository function `projray2hypersphere.m`)

2.2c testing that all coordinates are positive [63] within a tolerance[29]

2.2d repeating steps 2.2a-2.2c until an intersection is found or a stop condition is reached (see `nnMax` below).

For further information on barycentric coordinates and its applications and generalizations, see [63–65, 82–95].

Due to the large number of facets per point of a high-dimensional triangulation (approximately 2000 facets per vertex for a 50 000 point VFZ triangulation, or 1×10^8 total facets), some simplifications are made in order to determine intersections of prediction points with the mesh. If every edge length of every facet were equal, only facets connected to the first NN would need to be considered to find a proper intersection. However,
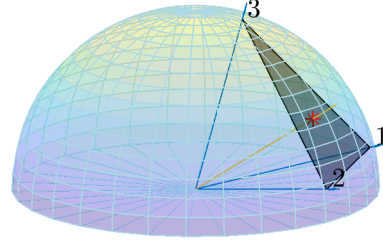
Figure B.3: A ray (red line) is linearly projected from the 2-sphere onto the hyperplane of a mesh facet (transparent black), shown as a red asterisk. The barycentric coordinates are computed as $\lambda_{i \in [1,3]} = \frac{1}{3}$. Because all barycentric coordinates are positive, it is determined that the projected point is an intersection with the mesh. Given vertex values of 8.183, 3.446, and 3.188 for vertices 1, 2, and 3, respectively, the interpolated value is calculated as 4.94 via Eq. (6).

since the VFZ-GBOs are randomly sampled, edge lengths of facets are non-uniform, and non-unity aspect-ratio facets exist (Figure B.2, Figure S7). If the facets have high-aspect ratios, the intersecting facets of prediction points can be far from the NNs mesh points relative to the prediction points (see Figure S7 inset), especially near the perimeter of a hyperspherical surface mesh. Rather than loop through every facet to find an intersection ($\sim 1 \times 10^8$ facets in a 50 000 VFZ-GBO mesh), the prediction point intersections are calculated by considering facets connected to up to some number of NN mesh vertices (`nnMax`) relative to each prediction point (in this work, `nnMax=10`). The NN mesh vertices relative to a prediction point are computed via the MATLAB built-in function `dsearchn` as in the NN approach (Section 2.3.4). The facet IDs of facets connected to these NNs are computed by calling built-in MATLAB function `find()`, as in `find(K==nn)`, where `K` is the triangulation from VFZ repository function `sphconvhulln.m` and `nn` is the ID of one of the NN mesh vertices.

Some prediction points will have no intersecting facet found. From our numerical testing, we determine that this non-intersection phenomenon occurs in two situations:

- high-aspect ratio facets (described above)

- prediction points that are positioned just outside the bounds of the mesh but within the bounds of the VFZ, due to the fact that the

---

[29]Two tolerances are used: one for the initial computation of barycentric coordinates by projecting onto the hypersphere to determine facet affiliation (`projtol=1e-4`) and a larger tolerance (`inttol=1e-2`) for computation of barycentric coordinates to determine interpolated values (Appendix B.3).

mesh is a piecewise linear approximation of a surface with a curved perimeter and that randomly sampled points typically do not fall on the true perimeter

In the first case, barycentric interpolation within high-aspect ratio facets may actually lead to worse interpolation error than a NN interpolation strategy due to influence by GBs far from the prediction point. In the second case, there is no true intersection between the prediction point and the mesh. Both issues can be addressed with the same strategy: we apply a NN approach (Section 2.3.4) when an intersecting facet is not found within `nnMax` NNs. In numerical tests, VFZ meshes composed of 388 and 50 000 vertices produced non-intersection rates of $(12.07 \pm 1.02)\,\%$ and $(0.68 \pm 0.11)\,\%$, respectively, over approximately 10 trials and using 10 000 prediction points for each trial.

Testing intersections for nearby facets is handled in the VFZ repository function `intersect_facet.m` and depends on the barycentric coordinate computations in `projray2hypersphere.m`.

B.3 Interpolation via Barycentric Coordinates

Once a mesh triangulation has been determined (Appendix B.1), barycentric coordinates are recomputed for a prediction point within the input mesh (Appendix B.2) using a somewhat larger tolerance; the interpolated value is found by taking the dot product of the prediction point's barycentric coordinates and the properties of the corresponding vertices of the intersecting facet via

$$v_{m,q} = \sum_{i=1}^{N} \lambda_{m,i} v_{m,i} \qquad (6)$$

where $\lambda_{m,i}$, $v_{m,q}$, $v_{m,i}$ and N, are the barycentric coordinates of the m-th prediction point, interpolated property at the m-th prediction point, property of the i-th vertex of the intersecting facet for the m-th prediction point, and number of vertices in a given facet ($N = 7$ for facets of the simplicial mesh on the degeneracy-free 6-sphere), respectively. Interpolation of many prediction points simultaneously can be accomplished by a simple,

vectorized approach via MATLAB built-in function `dot()` as used in VFZ repository function `interp_bary_fast.m`. This function assumes triangulation and weights have been precomputed. In other words, both input and prediction coordinates remain fixed, and only input property values change. If this is the case, barycentric interpolation of new points is incredibly fast. By contrast, if input coordinates change, the triangulation must be recomputed, and if prediction coordinates change, the intersecting facets must be recomputed. Both triangulation and finding intersecting facets are computationally demanding with respect to memory and runtime (Section 3.2).

Glossary

5DOF five degree-of-freedom 1, 2, 9, 11, 18

ANN artificial neural network 2, 15, 20

BRK Bulatov Reed Kumar 11, 13–20, 23

DOF degree of freedom 1, 21

FZ fundamental zone 3, 5, 9

GB grain boundary 1–3, 5–9, 11–15, 17–23, 28

GBC grain boundary character 2, 16, 18

GBE grain boundary energy 1, 2, 8, 11–23

GBO grain boundary octonion 1–11, 13, 14, 16, 17, 19, 20, 22, 24, 25

GPR Gaussian process regression 1, 3, 9–23

GPRM Gaussian process regression mixture 10, 11, 19, 21–23

IDW inverse-distance weighting 1, 2, 9–18, 23

MAE mean absolute error 2, 7, 8, 11, 12, 14–16, 19–22

MS molecular statics 1, 2, 11, 15, 19, 20, 22

NN nearest neighbor 1, 5, 6, 9–18, 23, 25, 27, 28

oSLERP octonion Spherical Linear Interpolation 2, 15, 18

RMSE root mean square error 1, 2, 7, 8, 11–16, 19–22

SEO symmetrically equivalent octonion 3–5, 8, 17, 18

SVD singular value decomposition 3, 10, 24–27

VFZ Voronoi fundamental zone 1–14, 16–21, 23–25, 27, 28

VFZ-GBO Voronoi fundamental zone grain boundary octonion 4–12, 14–19, 22–25, 27

References

[1] S. Jin, M. Huang, Y. Kwon, L. Zhang, B. W. Li, S. Oh, J. Dong, D. Luo, M. Biswal, B. V. Cunning, P. V. Bakharev, I. Moon, W. J. Yoo, D. C. Camacho-Mojica, Y. J. Kim, S. H. Lee, B. Wang, W. K. Seong, M. Saxena, F. Ding, H. J. Shin, R. S. Ruoff, Colossal grain growth yields single-crystal metal foils by contact-free annealing, Science 362 (2018) 1021–1025. doi:10.1126/science.aao3373.

[2] J. E. Brandenburg, L. A. Barrales-Mora, D. A. Molodov, On migration and faceting of low-angle grain boundaries: Experimental and computational study, Acta Materialia 77 (2014) 294–309. doi:10.1016/j.actamat.2014.06.006.

[3] Z. Huang, M. Bartels, R. Xu, M. Osterhoff, S. Kalbfleisch, M. Sprung, A. Suzuki, Y. Takahashi, T. N. Blanton, T. Salditt, J. Miao, Grain rotation and lattice deformation during photoinduced chemical reactions revealed by in situ X-ray nanodiffraction, Nature Materials 14 (2015) 691–695. doi:10.1038/nmat4311.

[4] Z. Trautt, Y. Mishin, Capillary-driven grain boundary motion and grain rotation in a tricrystal: A molecular dynamics study, Acta Materialia 65 (2014) 19–31. doi:10.1016/j.actamat.2013.11.059.

[5] H. Sharma, R. M. Huizenga, A. Bytchkov, J. Sietsma, S. E. Offerman, Observation of changing crystal orientations during grain coarsening, Acta Materialia 60 (2012) 229–237. doi:10.1016/j.actamat.2011.09.057.

[6] L. G. Ware, D. H. Suzuki, K. R. Wicker, Z. C. Cordero, Grain boundary plane manipulation in directionally solidified bicrystals and tricrystals, Scripta Materialia 152 (2018) 98–101. doi:10.1016/j.scriptamat.2018.03.047.

[7] J. Li, A. Oudriss, A. Metsue, J. Bouhattate, X. Feaugas, Anisotropy of hydrogen diffusion in nickel single crystals: The effects of self-stress and hydrogen concentration on diffusion, Scientific Reports 7 (2017) 45041. doi:10.1038/srep45041.

[8] A. Oudriss, J. Creus, J. Bouhattate, E. Conforto, C. Berziou, C. Savall, X. Feaugas, Grain size and grain-boundary effects on diffusion and trapping of hydrogen in pure nickel, Acta Materialia 60 (2012) 6814–6828. doi:10.1016/j.actamat.2012.09.004.

[9] A. Metsue, A. Oudriss, X. Feaugas, Hydrogen solubility and vacancy concentration in nickel single crystals at thermal equilibrium: New insights from statistical mechanics and ab initio calculations, Journal of Alloys and Compounds 656 (2016) 555–567. doi:10.1016/j.jallcom.2015.09.252.

[10] S. Huang, D. Chen, J. Song, D. L. McDowell, T. Zhu, Hydrogen embrittlement of grain boundaries in nickel: An atomistic study, npj Computational Materials 3 (2017) 1–8. doi:10.1038/s41524-017-0031-1.

[11] S. Xia, H. Li, T. G. Liu, B. X. Zhou, Appling grain boundary engineering to Alloy 690 tube for enhancing intergranular corrosion resistance, Journal of Nuclear Materials 416 (2011) 303–310. doi:10.1016/j.jnucmat.2011.06.017.

[12] M. J. Demkowicz, A threshold density of helium bubbles induces a ductile-to-brittle transition at a grain boundary in nickel, Journal of Nuclear Materials 533 (2020) 152118. doi:`10.1016/j.jnucmat.2020.152118`.

[13] J. P. Hanson, A. Bagri, J. Lind, P. Kenesei, R. M. Suter, S. Gradečak, M. J. Demkowicz, Crystallographic character of grain boundaries resistant to hydrogen-assisted fracture in Ni-base alloy 725, Nature Communications 9 (2018) 1–11. doi:`10.1038/s41467-018-05549-y`.

[14] S. Jothi, S. V. Merzlikin, T. N. Croft, J. Andersson, S. G. Brown, An investigation of micro-mechanisms in hydrogen induced cracking in nickel-based superalloy 718, Journal of Alloys and Compounds 664 (2016) 664–681. doi:`10.1016/j.jallcom.2016.01.033`.

[15] X. Zhou, D. Marchand, D. L. McDowell, T. Zhu, J. Song, Chemomechanical Origin of Hydrogen Trapping at Grain Boundaries in fcc Metals, Physical Review Letters 116 (2016) 1–33. doi:`10.1103/PhysRevLett.116.075502`.

[16] W. Huang, M. Shishehbor, N. Guarín-Zapata, N. D. Kirchhofer, J. Li, L. Cruz, T. Wang, S. Bhowmick, D. Stauffer, P. Manimunda, K. N. Bozhilov, R. Caldwell, P. Zavattieri, D. Kisailus, A natural impact-resistant bicontinuous composite nanoparticle coating, Nature Materials 19 (2020) 1236–1243. doi:`10.1038/s41563-020-0768-7`.

[17] Y. M. Wang, T. Voisin, J. T. McKeown, J. Ye, N. P. Calta, Z. Li, Z. Zeng, Y. Zhang, W. Chen, T. T. Roehling, R. T. Ott, M. K. Santala, P. J. Depond, M. J. Matthews, A. V. Hamza, T. Zhu, Additively manufactured hierarchical stainless steels with high strength and ductility, Nature Materials 17 (2018) 63–71. doi:`10.1038/nmat5021`.

[18] N. Y. C. Lin, M. Bierbaum, P. Schall, J. P. Sethna, I. Cohen, Measuring nonlinear stresses generated by defects in 3D colloidal crystals, Nature Materials 15 (2016) 1172–1176. doi:`10.1038/nmat4715`.

[19] D. Yin, C. Chen, M. Saito, K. Inoue, Y. Ikuhara, Ceramic phases with one-dimensional long-range order, Nature Materials 18 (2019) 19–23. doi:`10.1038/s41563-018-0240-0`.

[20] Y. Guan, W. Li, Y. Gong, G. Liu, X. Zhang, J. Chen, J. Gelb, W. Yun, Y. Xiong, Y. Tian, H. Wang, Analysis of the three-dimensional microstructure of a solid-oxide fuel cell anode using nano X-ray tomography, Journal of Power Sources 196 (2011) 1915–1919. doi:`10.1016/j.jpowsour.2010.09.059`.

[21] I. V. Vlassiouk, Y. Stehle, P. R. Pudasaini, R. R. Unocic, P. D. Rack, A. P. Baddorf, I. N. Ivanov, N. V. Lavrik, F. List, N. Gupta, K. V. Bets, B. I. Yakobson, S. N. Smirnov, Evolutionary selection growth of two-dimensional materials on polycrystalline substrates, Nature Materials 17 (2018) 318–322. doi:`10.1038/s41563-018-0019-3`.

[22] Y. Han, M.-Y. Li, G.-S. Jung, M. A. Marsalis, Z. Qin, M. J. Buehler, L.-J. Li, D. A. Muller, Sub-nanometre channels embedded in two-dimensional materials, Nature Materials 17 (2018) 129–133. doi:`10.1038/nmat5038`.

[23] J. Sun, J. Yu, Y. Guo, Q. Wang, Enhancing power factor of SnSe sheet with grain boundary by doping germanium or silicon, npj Computational Materials 6 (2020) 1–6. doi:`10.1038/s41524-020-00368-6`.

[24] O. K. Johnson, L. Li, M. J. Demkowicz, C. A. Schuh, Inferring grain boundary structure–property relations from effective property measurements, Journal of Materials Science 50 (2015) 6907–6919. doi:`10.1007/s10853-015-9241-4`.

[25] C.-C. Yang, A. Rollett, W. Mullins, Measuring relative grain boundary energies and mobilities in an aluminum foil from triple junction geometry, Scripta Materialia 44 (2001) 2735–2740. doi:`10.1016/S1359-6462(01)00960-5`.

[26] J. Zhang, W. Ludwig, Y. Zhang, H. H. B. Sørensen, D. J. Rowenhorst, A. Yamanaka, P. W. Voorhees, H. F. Poulsen, Grain boundary mobilities in polycrystals, Acta Materialia 191 (2020) 211–220. doi:10.1016/j.actamat.2020.03.044.

[27] J. Han, V. Vitek, D. J. Srolovitz, Grain-boundary metastability and its statistical properties, Acta Materialia 104 (2016) 259–273. doi:10.1016/j.actamat.2015.11.035.

[28] J. Wei, B. Feng, R. Ishikawa, T. Yokoi, K. Matsunaga, N. Shibata, Y. Ikuhara, Direct imaging of atomistic grain boundary migration, Nature Materials (2021). doi:10.1038/s41563-020-00879-z.

[29] R. Bostanabad, A. T. Bui, W. Xie, D. W. Apley, W. Chen, Stochastic microstructure characterization and reconstruction via supervised learning, Acta Materialia 103 (2016). doi:10.1016/j.actamat.2015.09.044.

[30] E. R. Homer, High-throughput simulations for insight into grain boundary structure-property relationships and other complex microstructural phenomena, Computational Materials Science 161 (2019) 244–254. doi:10.1016/j.commatsci.2019.01.041.

[31] S. Jothi, T. N. Croft, S. G. Brown, Multi-scale multiphysics model for hydrogen embrittlement in polycrystalline nickel, Journal of Alloys and Compounds 645 (2015) S500–S504. doi:10.1016/j.jallcom.2014.12.073.

[32] H. Pirgazi, On the alignment of 3D EBSD data collected by serial sectioning technique, Materials Characterization 152 (2019) 223–229. doi:10.1016/j.matchar.2019.04.026.

[33] H. Pirgazi, K. Glowinski, A. Morawiec, L. A. Kestens, Three-dimensional characterization of grain boundaries in pure nickel by serial sectioning via mechanical polishing, Journal of Applied Crystallography 48 (2015) 1672–1678. doi:10.1107/S1600576715017616.

[34] A. Speidel, R. Su, J. Mitchell-Smith, P. Dryburgh, I. Bisterov, D. Pieris, W. Li, R. Patel, M. Clark, A. T. Clare, Crystallographic texture can be rapidly determined by electrochemical surface analytics, Acta Materialia 159 (2018) 89–101. doi:10.1016/J.ACTAMAT.2018.07.059.

[35] H. Zheng, X. G. Li, R. Tran, C. Chen, M. Horton, D. Winston, K. A. Persson, S. P. Ong, Grain boundary properties of elemental metals, Acta Materialia 186 (2020) 40–49. doi:10.1016/j.actamat.2019.12.030. arXiv:1907.08905.

[36] R. Keinan, H. Bale, N. Gueninchault, E. Lauridsen, A. Shahani, Integrated imaging in three dimensions: Providing a new lens on grain boundaries, particles, and their correlations in polycrystalline silicon, Acta Materialia 148 (2018) 225–234. doi:10.1016/J.ACTAMAT.2018.01.045.

[37] M. Seita, M. Volpi, S. Patala, I. McCue, C. A. Schuh, M. V. Diamanti, J. Erlebacher, M. J. Demkowicz, A high-throughput technique for determining grain boundary character nondestructively in microstructures with through-thickness grains, Npj Computational Materials 2 (2016) 16016. URL: http://dx.doi.org/10.1038/npjcompumats.2016.16.

[38] B. Winiarski, A. Gholinia, K. Mingard, M. Gee, G. Thompson, P. Withers, Broad ion beam serial section tomography, Ultramicroscopy 172 (2017) 52–64. doi:10.1016/j.ultramic.2016.10.014.

[39] H. K. Kim, W. S. Ko, H. J. Lee, S. G. Kim, B. J. Lee, An identification scheme of grain boundaries and construction of a grain boundary energy database, Scripta Materialia 64 (2011) 1152–1155. doi:10.1016/j.scriptamat.2011.03.020.

[40] S. Li, L. Yang, C. Lai, Atomistic simulations of energies for arbitrary grain boundaries. Part I: Model and validation, Compu-

tational Materials Science 161 (2019) 330–338. doi:10.1016/j.commatsci.2019.02.003.

[41] J. Li, S. J. Dillon, G. S. Rohrer, Relative grain boundary area and energy distributions in nickel, Acta Materialia 57 (2009) 4304–4311. doi:10.1016/j.actamat.2009.06.004.

[42] D. L. Olmsted, E. A. Holm, S. M. Foiles, Survey of computed grain boundary properties in face-centered cubic metals-II: Grain boundary mobility, Acta Materialia 57 (2009) 3704–3713. doi:10.1016/j.actamat.2009.04.015.

[43] D. L. Olmsted, S. M. Foiles, E. A. Holm, Survey of computed grain boundary properties in face-centered cubic metals: I. Grain boundary energy, Acta Materialia 57 (2009) 3694–3703. doi:10.1016/j.actamat.2009.04.007.

[44] V. Randle, G. S. Rohrer, H. M. Miller, M. Coleman, G. T. Owen, Five-parameter grain boundary distribution of commercially grain boundary engineered nickel and copper, Acta Materialia 56 (2008) 2363–2373. doi:10.1016/j.actamat.2008.01.039.

[45] D. M. Saylor, A. Morawiec, B. L. Adams, G. S. Rohrer, Misorientation dependence of the grain boundary energy in magnesia, Interface Science 8 (2000) 131–140. doi:10.1023/A:1008764219575.

[46] D. M. Saylor, A. Morawiec, G. S. Rohrer, The relative free energies of grain boundaries in magnesia as a function of five macroscopic parameters, Acta Materialia 51 (2003) 3675–3686. doi:10.1016/S1359-6454(03)00182-4.

[47] L. Yang, C. Lai, S. Li, Atomistic simulations of energies for arbitrary grain boundaries. Part II: Statistical analysis of energies for tilt and twist grain boundaries, Computational Materials Science 162 (2019) 268–276. doi:10.1016/j.commatsci.2019.03.010.

[48] S. J. Dillon, G. S. Rohrer, Characterization of the grain-boundary character and energy distributions of yttria using automated serial sectioning and ebsd in the FIB, Journal of the American Ceramic Society 92 (2009) 1580–1585. doi:10.1111/j.1551-2916.2009.03064.x.

[49] S. E. Restrepo, S. T. Giraldo, B. J. Thijsse, Using artificial neural networks to predict grain boundary energies, Computational Materials Science 86 (2014) 170–173. doi:10.1016/j.commatsci.2014.01.039.

[50] M. Guziewski, D. Montes de Oca Zapiain, R. Dingreville, S. P. Coleman, Microscopic and Macroscopic Characterization of Grain Boundary Energy and Strength in Silicon Carbide via Machine-Learning Techniques, ACS Applied Materials & Interfaces 13 (2021) 3311–3324. doi:10.1021/acsami.0c15980.

[51] C. Hu, Y. Zuo, C. Chen, S. Ping Ong, J. Luo, Genetic algorithm-guided deep learning of grain boundary diagrams: Addressing the challenge of five degrees of freedom, Materials Today 38 (2020) 49–57. doi:10.1016/j.mattod.2020.03.004. arXiv:2002.10632.

[52] T. Francis, I. Chesser, S. Singh, E. A. Holm, M. De Graef, A geodesic octonion metric for grain boundaries, Acta Materialia 166 (2019) 135–147. doi:10.1016/j.actamat.2018.12.034.

[53] I. Chesser, T. Francis, M. De Graef, E. Holm, Learning the grain boundary manifold: Tools for visualizing and fitting grain boundary properties, Acta Materialia 195 (2020) 209–218. doi:10.1016/j.actamat.2020.05.024.

[54] A. Morawiec, On distances between grain interfaces in macroscopic parameter space, Acta Materialia 181 (2019) 399–407. doi:10.1016/j.actamat.2019.09.032.

[55] C. B. Barber, D. P. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Transactions on Mathematical Software 22 (1996) 469–483. doi:10.1145/235815.235821.

[56] A. Heinz, P. Neumann, Representation of orientation and disorientation data for cubic, hexagonal, tetragonal and orthorhombic crystals, Acta Crystallographica Section A 47 (1991) 780–789. doi:10.1107/S0108767391006864.

[57] H. Grimmer, A unique description of the relative orientation of neighbouring grains, Acta Crystallographica Section A 36 (1980) 382–389. doi:10.1107/S0567739480000861.

[58] S. Patala, C. A. Schuh, Symmetries in the representation of grain boundary-plane distributions, Philosophical Magazine 93 (2013) 524–573. doi:10.1080/14786435.2012.722700.

[59] B. Luong, Voronoi Sphere, MATLAB Central File Exchange, 2020. URL: https://www.mathworks.com/matlabcentral/fileexchange/40989-voronoi-sphere.

[60] S. Baird, O. Johnson, Five Degree-of-Freedom (5DOF) Interpolation, 2020. URL: github.com/sgbaird-5dof/interp.

[61] E. R. Homer, S. Patala, J. L. Priedeman, Grain Boundary Plane Orientation Fundamental Zones and Structure-Property Relationships, Scientific Reports 5 (2015) 1–13. doi:10.1038/srep15476.

[62] S. Singh, M. De Graef, Orientation sampling for dictionary-based diffraction pattern indexing methods, Modelling and Simulation in Materials Science and Engineering 24 (2016). doi:10.1088/0965-0393/24/8/085013.

[63] T. Langer, A. Belyaev, H.-P. Seidel, Spherical barycentric coordinates, Proceedings of the fourth Eurographics symposium on Geometry processing (2006) 81–88. URL: http://portal.acm.org/citation.cfm?id=1281957.1281968.

[64] M. Floater, Generalized barycentric coordinates and applications, Acta Numerica 24 (2015) 161–214. doi:10.1017/S09624929. arXiv:1711.05337v1.

[65] M. Meyer, A. Barr, H. Lee, M. Desbrun, Generalized Barycentric Coordinates on Irregular Polygons, Journal of Graphics Tools 7 (2002) 13–22. doi:10.1080/10867651.2002.10487551.

[66] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning, MIT Press, Cambridge, Mass, 2006.

[67] A. Tovar, Inverse distance weight function, MATLAB Central File Exchange, 2020. URL: https://www.mathworks.com/matlabcentral/fileexchange/46350-inverse-distance-weight-function.

[68] H.-K. Kim, S. G. Kim, W. Dong, I. Steinbach, B.-J. Lee, Phase-field modeling for 3D grain growth based on a grain boundary energy database, Modelling and Simulation in Materials Science and Engineering 22 (2014) 034004. doi:10.1088/0965-0393/22/3/034004.

[69] I. Chesser, GB Octonion Code, 2019. URL: https://github.com/ichesser/GB_octonion_code.

[70] V. V. Bulatov, B. W. Reed, M. Kumar, Grain boundary energy function for fcc metals, Acta Materialia 65 (2014) 161–175. doi:10.1016/j.actamat.2013.10.057.

[71] G. Bean, Hexscatter, MATLAB Central File Exchange, 2020. URL: https://www.mathworks.com/matlabcentral/fileexchange/45639-hexscatter-m.

[72] E. D. Dolan, J. J. More, T. S. Munson, Benchmarking Optimization Software with COPS 3.0, Technical Report, Argonne National Laboratory (ANL), United States, 2004. doi:10.2172/834714.

[73] MATLAB Optimization Toolbox, Constrained Electrostatic Nonlinear Optimization, Problem-Based, 2020. URL: https://www.mathworks.com/help/optim/ug/constrained-electrostatic-problem-based-optimization.html.

[74] Y. F. Shen, X. Zhong, H. Liu, R. M. Suter, A. Morawiec, G. S. Rohrer, Determining grain boundary energies from triple junction geometries without discretizing the five-parameter space, Acta Materialia 166 (2019) 126–134. doi:10.1016/j.actamat.2018.12.022.

[75] M. De Graef, EMSoft, 2020. doi:10.5281/zenodo.3489720.

[76] A. Dimokrati, Y. Le Bouar, M. Benyoucef, A. Finel, S-PFM model for ideal grain growth, Acta Materialia 201 (2020) 147–157. doi:10.1016/j.actamat.2020.09.073.

[77] E. Miyoshi, T. Takaki, S. Sakane, M. Ohno, Y. Shibuta, T. Aoki, Large-scale phase-field study of anisotropic grain growth: Effects of misorientation-dependent grain boundary energy and mobility, Computational Materials Science 186 (2021) 109992. doi:10.1016/j.commatsci.2020.109992.

[78] D. Rowenhorst, A. D. Rollett, G. S. Rohrer, M. Groeber, M. Jackson, P. J. Konijnenberg, M. De Graef, Consistent representations of and conversions between 3D rotations, Modelling and Simulation in Materials Science and Engineering 23 (2015) 083501. doi:10.1088/0965-0393/23/8/083501.

[79] R. Connor, L. Vadicamo, F. Rabitti, High-dimensional simplexes for supermetric search, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10609 LNCS (2017) 96–109. doi:10.1007/978-3-319-68474-1_7. arXiv:1707.08370.

[80] J. D. Boissonnat, R. Dyer, A. Ghosh, S. Y. Oudot, Only distances are required to reconstruct submanifolds, Computational Geometry: Theory and Applications 66 (2017) 32–67. doi:10.1016/j.comgeo.2017.08.001. arXiv:1410.7012.

[81] T. Anatoliy, Check if ray intersects internals of D-facet, Mathematics Stack Exchange, 2015. URL: https://math.stackexchange.com/q/1256236.

[82] V. Skala, Robust Barycentric Coordinates Computation of the Closest Point to a Hyperplane in Eñ, Proceedings of the 2013 International Conference on Applies Mathematics and Computational Methods in Engineering (2013) 239–244.

[83] D. Anisimov, C. Deng, K. Hormann, Subdividing barycentric coordinates, Computer Aided Geometric Design 43 (2016) 172–185. doi:10.1016/j.cagd.2016.02.005.

[84] M. Budninskiy, B. Liu, Y. Tong, M. Desbrun, Power coordinates: A geometric construction of barycentric coordinates on convex polytopes, ACM Transactions on Graphics 35 (2016). doi:10.1145/2980179.2982441.

[85] R. Dyer, G. Vegter, M. Wintraecken, Barycentric coordinate neighbourhoods in Riemannian manifolds, arXiv (2016). URL: http://arxiv.org/abs/1606.01585. arXiv:1606.01585.

[86] M. S. Floater, J. Kosinka, On the injectivity of Wachspress and mean value mappings between convex polygons, Advances in Computational Mathematics 32 (2010) 163–174. doi:10.1007/s10444-008-9098-z.

[87] K. Hormann, J. Kosinka, Discretizing Wachspress kernels is safe, Computer Aided Geometric Design 52-53 (2017) 126–134. doi:10.1016/j.cagd.2017.02.015.

[88] K. Hormann, N. Sukumar, Maximum entropy coordinates for arbitrary polytopes, Eurographics Symposium on Geometry Processing 27 (2008) 1513–1520.

[89] T. Langer, H. P. Seidel, Higher order barycentric coordinates, Computer Graphics Forum 27 (2008) 459–466. doi:10.1111/j.1467-8659.2008.01143.x.

[90] K. Lei, D. Qi, X. Tian, A New Coordinate System for Constructing Spherical Grid Systems, Applied Sciences 10 (2020) 655. doi:10.3390/app10020655.

[91] P. S. Peixoto, S. R. Barros, On vector field reconstructions for semi-Lagrangian transport methods on geodesic staggered grids, Journal of Computational Physics 273 (2014) 185–211. doi:10.1016/j.jcp.2014.04.043.

[92] P. Pihajoki, M. Mannerkoski, P. H. Johansson, Barycentric interpolation on Riemannian and semi-Riemannian spaces, Monthly Notices of the Royal Astronomical Society 489 (2019) 4161–4169. doi:10.1093/mnras/stz2447. arXiv:1907.09487.

[93] R. M. Rustamov, Barycentric coordinates on surfaces, Computer Graphics Forum 29 (2010) 1507–1516. doi:10.1111/j.1467-8659.2010.01759.x.

[94] J. Tao, B. Deng, J. Zhang, A fast numerical solver for local barycentric coordinates, Computer Aided Geometric Design 70 (2019) 46–58. doi:10.1016/j.cagd.2019.04.006.

[95] J. Warren, S. Schaefer, A. N. Hirani, M. Desbrun, Barycentric coordinates for convex sets, Advances in Computational Mathematics 27 (2007) 319–338. doi:10.1007/s10444-005-9008-6.