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

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*

Contents

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

S0.1. Use of Interpolation Function

To facilitate easy application of the presented methods, a vectorized, parallelized, MATLAB implementation, `interp5DOF.m`, is made available in the Voronoi fundamental zone (VFZ) repository [1] with similar input/output structure to that of built-in MATLAB interpolation functions (e.g. `scatteredInterpolant()`, `griddatan()`). A typical function call is as follows: `ypred = interp5DOF(qm,nA,y,qm2,nA2,method)`. The argument `y` is a vector of known property values corresponding to the GBs defined by (`qm`,`nA`), which respectively denote pairs of GB misorientation quaternions and boundary plane normals. The result, `ypred`, is a vector of predicted/interpolated property values corresponding to the prediction grain boundaries (GBs) defined by (`qm2`,`nA2`).

Internally, these are converted to grain boundary octonions (GBOs) and interpolation is performed using the selected `method`. For the validation function, these can be compared to the true grain boundary energys (GBEs) `ytrue`. The methods used in this work are `'pbary'`, `'gpr'`, `'idw'`, and `'nn'`, corresponding to planar barycentric, Gaussian process regression (GPR), inverse-distance weighting, and nearest neighbor (NN) interpolation, respectively. A placeholder template with instructions for implementing additional interpolation schemes is also provided in `interp5DOF.m`. See Francis et al. [2] and `five2oct.m` [1] treatments of conversions to GBO coordinates in the passive and active senses, respectively (Appendix A).

S0.2. 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, to generate random GBOs from five degree-of-freedom 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 [3] (`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[1]. 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 Voronoi fundamental zone grain boundary octonions (VFZ-GBOs)) via `osym=get_octpairs(o)`. A default reference GBO[2] is used for these calculations, unless specified by the user. We use the active 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.

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

[2] 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 [4].

S1. Euclidean and Arc Length Distances

In addition to enabling us to leverage the machinery of efficient and established algorithms, the choice of using Euclidean distance as opposed to hyperspherical arclength can be justified by the following observations:

- The minimum Euclidean distance symmetrically equivalent octonion (SEO) will be the same as the minimum arc length distance SEO because d_{S} is a monotonically increasing function of d_{E}, for $d_{\mathrm{S}}(d_{\mathrm{E}}) \in [0, \pi]$ (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 $d_{\mathrm{E}} \simeq d_{\mathrm{S}}$ holds to very high accuracy[3] 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}.

The close correlation between Euclidean and arc length distances in the VFZ-GBO sense is shown in Figure S1 using pairwise distances of 10 000 VFZ-GBOs. This justifies our use of Euclidean distance as an approximation of hyperspherical arc length (and by extension, that a scaled Euclidean distance approximates a non-symmetrized GBO distance, see Eqs. (1)–(3) of the main paper). However, comparison with the original GBO metric [2] gives overestimation for some boundaries. This is an inherent feature of the VFZ framework that can be addressed via use of the ensemble methods described in Section 2.1.3 (see also Figures 2 and 3).

Additionally, the use of an isometry equivalence relationship in Morawiec [4] in a non-VFZ sense results in identical distance results within numerical tolerance (Figure S2).

S2. Computational Complexity of VFZ vs. GBO Distances

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. [2], 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. [2]). Thus, the total number of SEO computations will be $4N_p^4L^2$. However, it is possible to fix a single GB in the GB pair and still obtain accurate[4] due to isometry equivalence (see Section 7 of [4] 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[5] 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[6] will be $O(N_p^2L)$, which represents a dramatic reduction compared to the traditional approach.

[3] 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.

[4] Compared with the pairwise distance matrix of the 388 Olmsted GBs, we obtained a root mean square error (RMSE) of $1.6566 \times 10^{-7\circ}$ for this computation which completed in 133 s using 6 cores (see `get_pd_fix.m`)

[5] This is 8 instead of 4 because the simplifying assumption that only two of the four double cover cases need to be considered [2] 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`).

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

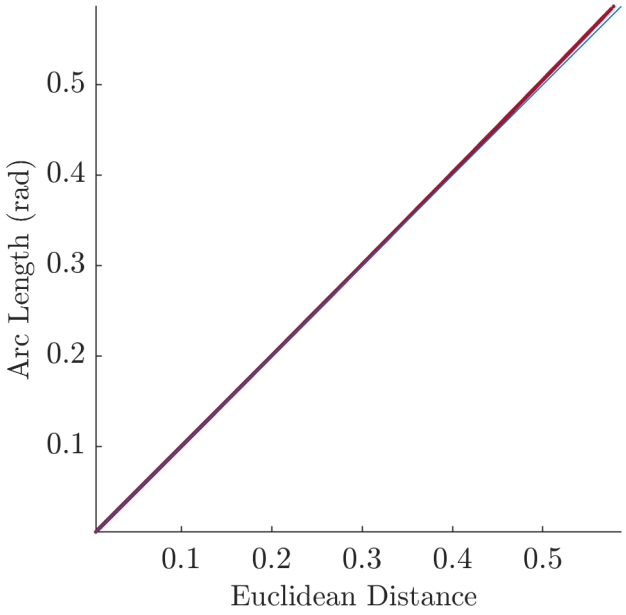

Figure S1: Parity plot of 8D Cartesian hyperspherical arc length vs. 8D Cartesian Euclidean distance for pairwise distances in a $(m\bar{3}m)$ symmetrized set of 10 000 randomly sampled VFZ-GBOs. The max arc length is approximately 0.58 rad, indicating a max GBO distance of approximately 1.16 rad or 66.5° between any two points in the VFZ. The close correlation between arc length and Euclidean distance supports the validity of using Euclidean distance instead of arc length in the interpolation methods. This is *separate* from the correlation between VFZ-GBO Euclidean or arc length distances with the traditional GBO distance [5].

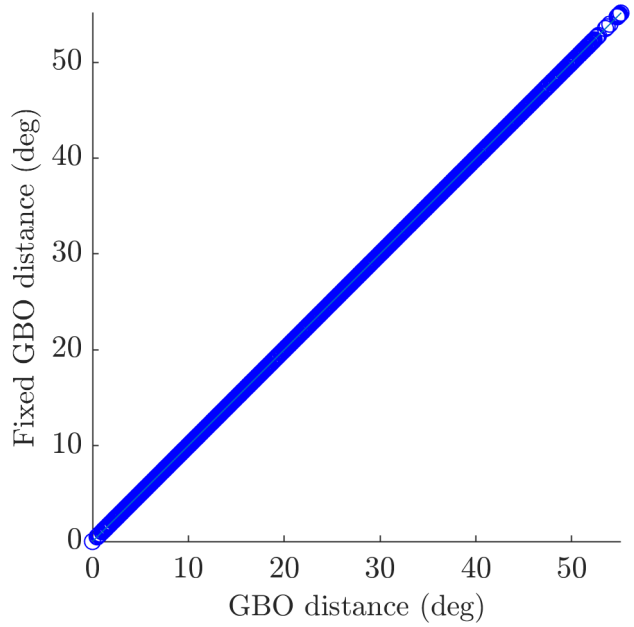

Figure S2: GB distances calculated with one GBO fixed vs. the traditional calculations in Chesser et al. [5] show that the isometry equivalence discussed in Morawiec [4] applies to GBOs. The pairwise-distance matrix for the Olmsted GBs supplied in [6] was used.

S3. Additional Interpolation Results

S3.1. Smaller Set Sizes of Input GBs

Interpolation results for 388 and 10 000 GBs are given in Figure S3 and Figure S4, respectively.

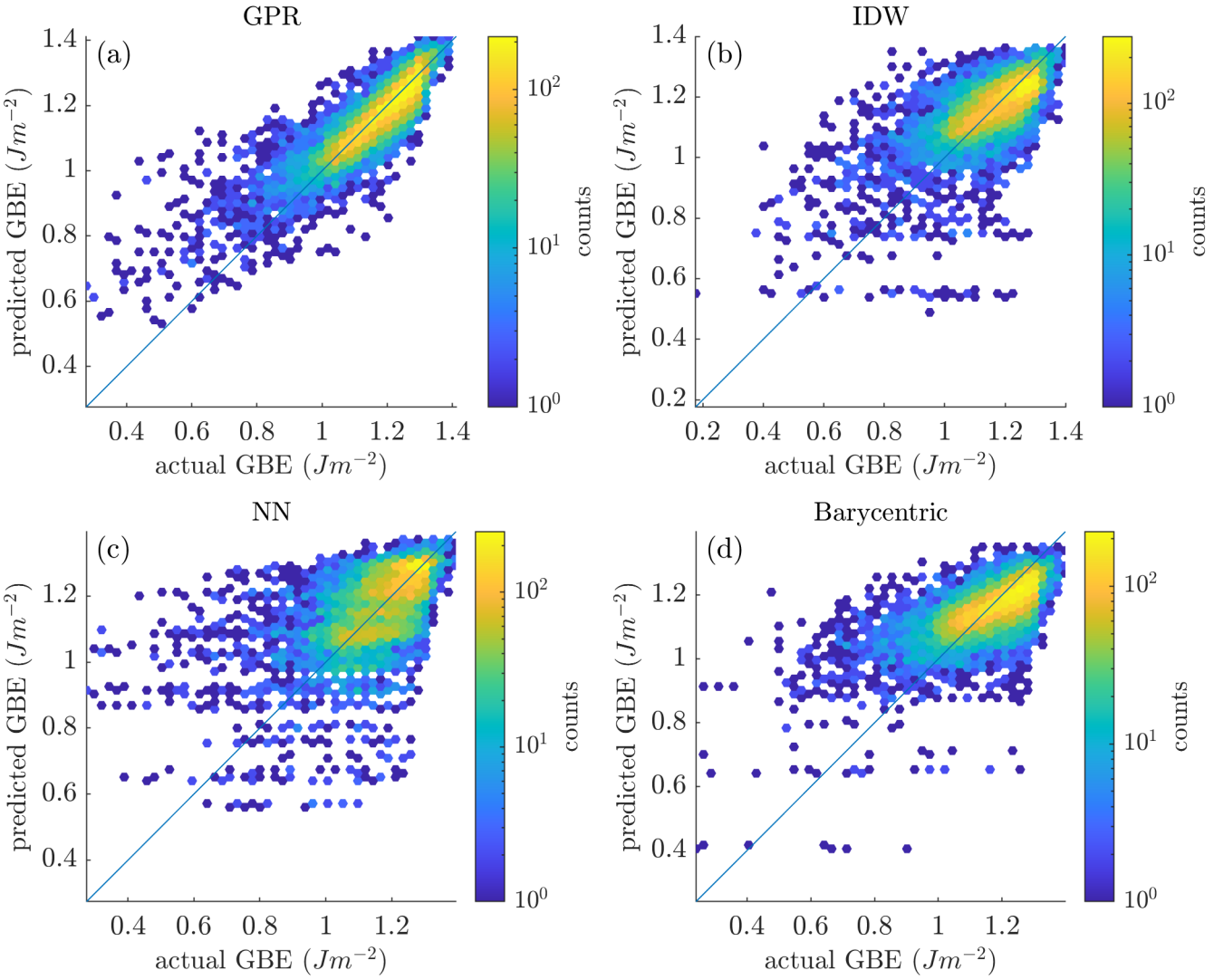

Figure S3: Hexagonally binned parity plots for 388 input and 10 000 prediction GBOs formed via pairs of a random cubochorically sampled quaternion and a spherically sampled random boundary plane normal. Interpolation via (a) GPR, (b) inverse-distance weighting, (c) NN, and (d) barycentric coordinates. Bulatov Reed Kumar GBE function for face-centered cubic Ni [7] was used as the test function.

S3.2. Octonions used for 1D Arcs

The octonions used in Figures 6 and 8 are given in Tables S1 and S2, respectively.

S4. Mitigating 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

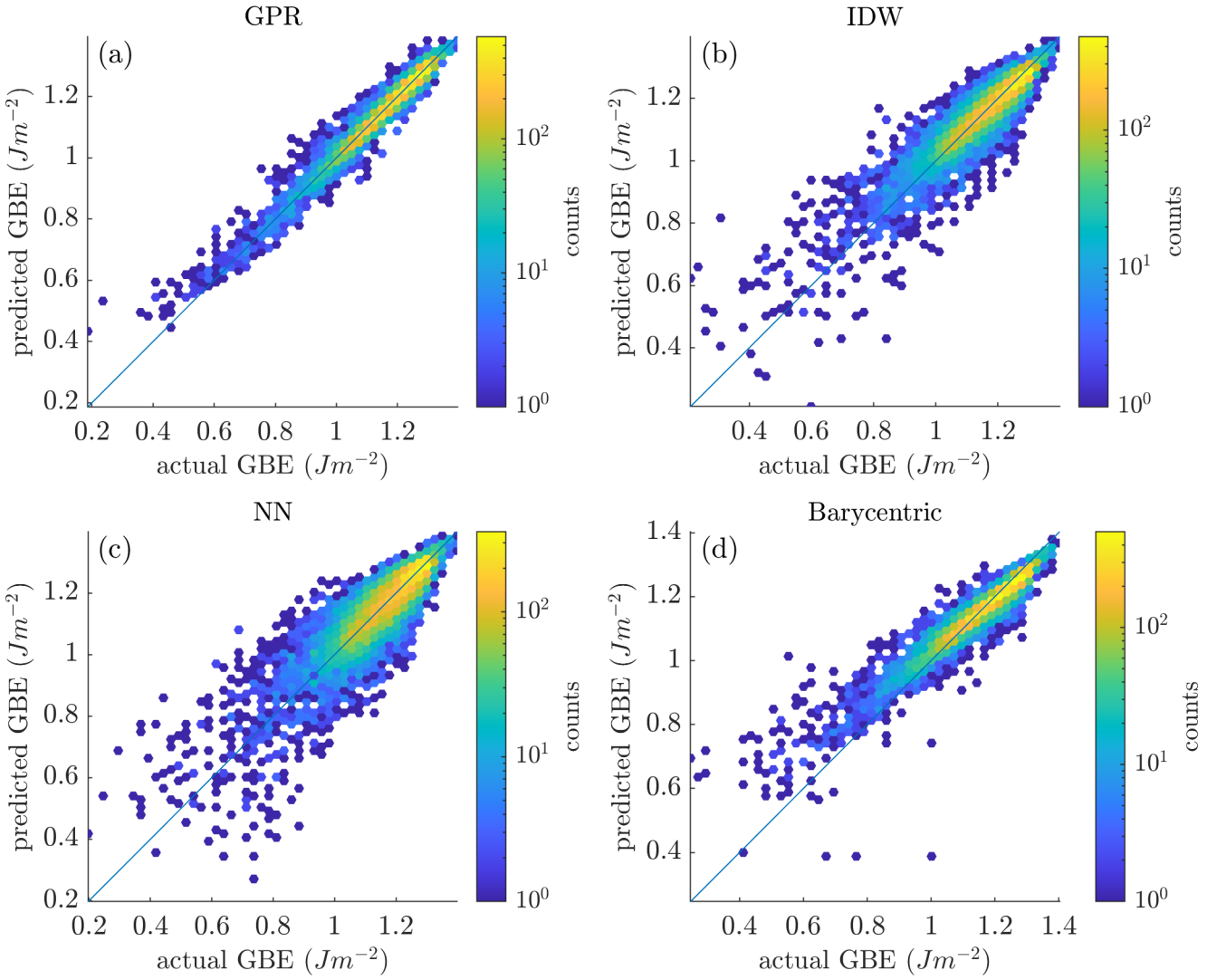

Figure S4: Hexagonally binned parity plots for 10 000 input and 10 000 prediction GBOs formed via pairs of a random cubochorically sampled quaternion and a spherically sampled random boundary plane normal. Interpolation via (a) GPR, (b) inverse-distance weighting, (c) NN, and (d) barycentric coordinates. Bulatov Reed Kumar GBE function for face-centered cubic Ni [7] was used as the test function.

between uncorrelated GBs. Because only overestimation relative to traditional GBO distances exist in this work (as shown in Figure 2), 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 when the point set size is large, 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 2. Additionally, Figure 3 shows that the error between scaled Euclidean distance and the traditional GBO metric decreases rapidly as the number of ensemble VFZ-GBO com-

Table S1: Approximate coordinates of VFZ-GBOs A and B used for the interpolation in Figure 6. 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

Table S2: Approximate coordinates of VFZ-GBOs A and B used for the molecular statics Fe simulation dataset interpolation in Figure 8. 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

ponents increases. This confirms that employing a small ensemble of VFZ-GBO sets results in significant improvement to the Euclidean distance approximation (Figures 2 and 3) 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 ∼10 via a loop-based implementation. For a symmetrized 50 000 × 50 000 pairwise distance matrix, this results in a runtime of approximately 1 CPU hour instead of ∼7 CPU minutes for a single VFZ. However, this is still much faster than the original GBO approach used in [5], 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 [4]). Additionally, it may be worthwhile to make the distance calculations GPU-compatible for further speed-up.

S5. Ensemble Interpolation Results

Ensemble interpolation is a classic technique that can be used to enhance predictive performance of models. Here we describe our methods (Section S5.1), results (Section S5.2), and the potential of integrating ensemble interpolation with a Gaussian process regression mixture scheme (Section S5.3).

S5.1. Methods

VFZ-GBO ensemble[7] interpolation occurs by:

1. generating multiple reference GBOs to define multiple VFZs
2. obtaining multiple VFZ-GBO representations for a set of GBs based on the various reference GBOs
3. performing an interpolation (e.g. GPR) for each of the representations
4. homogenizing the ensemble of models (e.g. by taking the mean or median of the various models)

[7] Ours is a "bagging"-esque ensemble scheme because the same interpolation method (GPR) is used but with different representations for the input data.

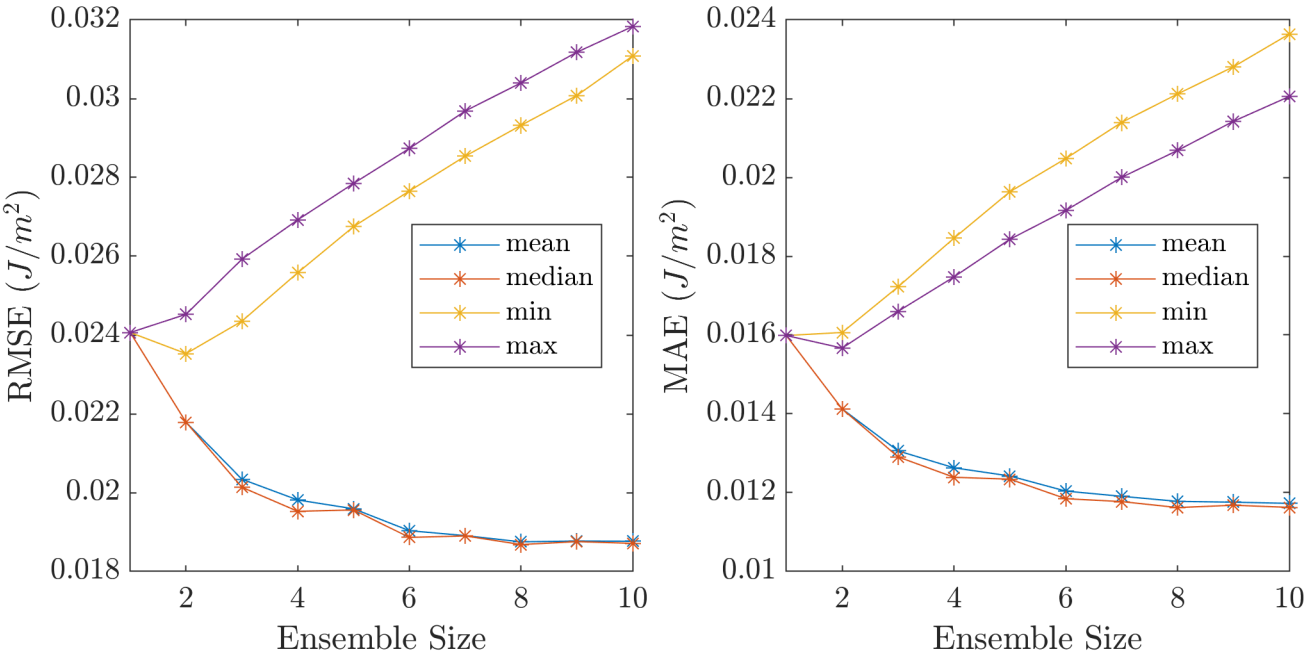

Figure S5: (a) RMSE and (b) mean absolute error vs. ensemble size for mean, median, minimum, and maximum homogenization functions. A GPR model with 50 000 input and 10 000 prediction VFZ-GBOs was used.

S5.2. Results

Use of an ensemble interpolation scheme decreases interpolation error for a GPR model with 50 000 input and 10 000 prediction VFZ-GBOs. By using an ensemble size of 10 (i.e. 10 GPR models each with different reference GBOs and therefore different VFZs), RMSE and mean absolute error decreased from 0.0241 J m^{-2} and 0.0160 J m^{-2} to 0.0187 J m^{-2} and 0.0116 J m^{-2}, respectively, using the median homogenization function (Figure S5).

Figure S6 shows the hexagonally binned parity plots for predictions made using the mean, median, minimum, and maximum predicted values over an ensemble of 10 VFZs. Qualitatively, the ensemble mean and ensemble median parity plots look similar to those from the main text (Figure 4), though the distributions of the ensemble scheme are somewhat tighter. The ensemble minimum produces better predictions of low GBE than any of the other models, but underestimates high GBE as expected. Naturally, the ensemble maximum overestimates in general. Diminishing returns manifest in Figure S5 for mean and median homogenizations. This is to be expected because the original GBO distances [2] are well-approximated using an ensemble size of 10 (Figure 2c and Figure 3).

S5.3. Possibility: Combining Ensemble with Gaussian Process Regression Mixture

A scheme which preferentially favors the ensemble minimum for low GBE predictions and defaults to ensemble mean or median for all other GBEs may produce even better results across the full range of GBEs. For example, this could be accomplished by combining the ensemble scheme described here with the GPR mixture model described in Section S6.4.

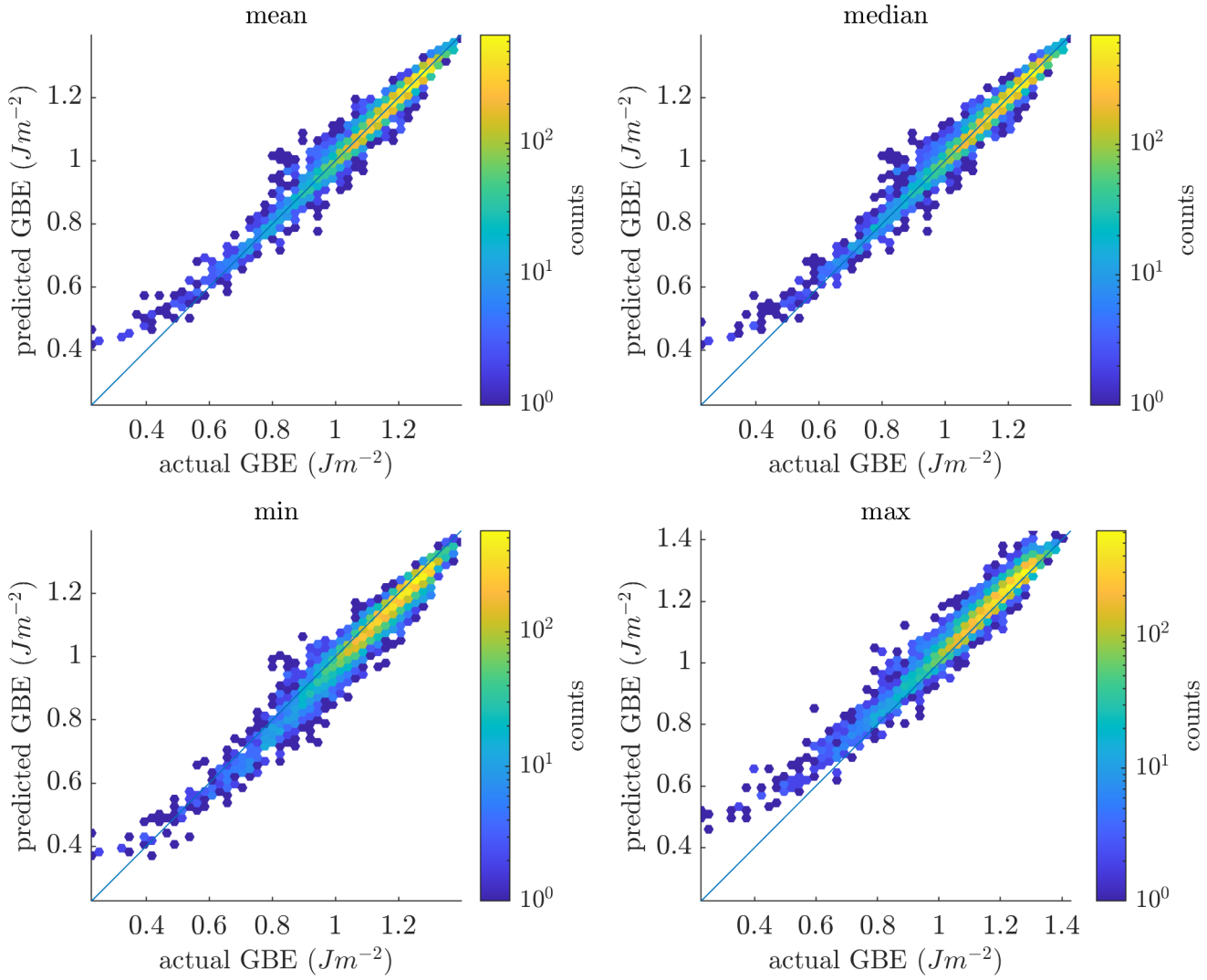

Figure S6: Hexagonally binned parity plots for (a) mean, (b) median, (c) minimum, and (d) maximum ensemble homogenization functions. A GPR model with 50 000 input and 10 000 prediction VFZ-GBOs was used.

S6. Literature Datasets

S6.1. Gaussian Process Regression for Fe Simulation Dataset

The Fe simulation data is obtained from [8] rather than [9] due to a mistake in the earlier dataset file[8]. GBs with a GBE less than 0.01 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"[9], and only the average GBE (and a single GB) within each degenerate set was retained.

[8] We were informed of the error during an email discussion with the corresponding author of [8].

[9] A degenerate "set" is distinct from a VFZ-GBOs "set". This sorting occurs via `avgrepeats.m` with `avgfn='min'`.

S6.2. Gaussian Process Regression for Ni Simulation Dataset

We use the GBO representations[10] [5] of GBs from [10], importing and converting them to the active sense by taking the quaternion inverse of each of the GBOs' quaternions. We take GBE values[11], and use a GPR model (Section 2.2.2).

S6.3. Gaussian Process Regression Mixture for Fe Simulation Dataset

Separate from the four main methods analyzed in this work, a Gaussian process regression mixture model is developed to better predict low GBE using the non-uniformly distributed, noisy, Fe simulation dataset described in Section S6.1. An exponential rather than a squared exponential kernel was used for the subset GPR model (Section S6.4) to accommodate sharper transitions to better approximate low GBEs.

S6.4. Details of Gaussian Process Regression Mixture

A GPR mixing model is developed to accommodate the non-uniformly distributed, noisy Fe simulation data [8] and better predict low GBE. The code implementation is given in `gprmix.m` and `gprmix_test.m` of the VFZ repository [1].

As shown in Figure S7a, prediction using the standard approach of the main document (termed the ϵ_1 model) overestimates low GBEs for this dataset. By training the model on only GBs with a GBE less than $1.2\,\mathrm{J\,m^{-2}}$ (termed the ϵ_2 model) and by using an exponential (`KernelFunction='exponential'`) rather than a squared exponential kernel, prediction of low GBEs improves, but naturally underestimation occurs for higher GBEs (Figure S7b).

A combined, disjoint model (Figure S7c) is taken (ϵ_3) by replacing ϵ_1 GBE predictions for GBs with GBE less than $1.1\,\mathrm{J\,m^{-2}}$ with the corresponding ϵ_2 predictions. Finally, a weighted average (Eq. (S1)) is taken according to:

$$\epsilon_{mix} = f\epsilon_1 + (f-1)\epsilon_2 \tag{S1}$$

where ϵ_1 and ϵ_2 represent the standard GPR model and the GPR model trained on the subset of GBs with a GBE less than $1.2\,\mathrm{J\,m^{-2}}$, respectively, and f is the sigmoid mixing fraction given by:

$$f = \frac{1}{e^{-m(\epsilon_3 - b)} + 1} \tag{S2}$$

and shown in Figure S8 with $m = 30$ and $b = 1.1\,\mathrm{J\,m^{-2}}$, as used in this work. Larger values of m yield a steeper sigmoid function and larger values of b shift the sigmoid function further to the right. Specific values for m and b were chosen by visual inspection and trial and error. This results in a GPR mixing model which better predicts low GBEs while retaining overall predictive accuracy (Figure S7d).

Uncertainty of the GPR mixing model is similarly obtained by taking a weighted average of the uncertainties of each model according to:

$$\sigma_{mix} = f\sigma_1 + (f-1)\sigma_2 \tag{S3}$$

where σ_1 and σ_2 are the corresponding uncertainties of ϵ_1 and ϵ_2, respectively, and f is given by Eq. (S2).

[10] Contained in `'olm_octonion_list.txt'` from Chesser [6].
[11] Contained in first column of `'olm_properties.txt'` from Chesser [6].

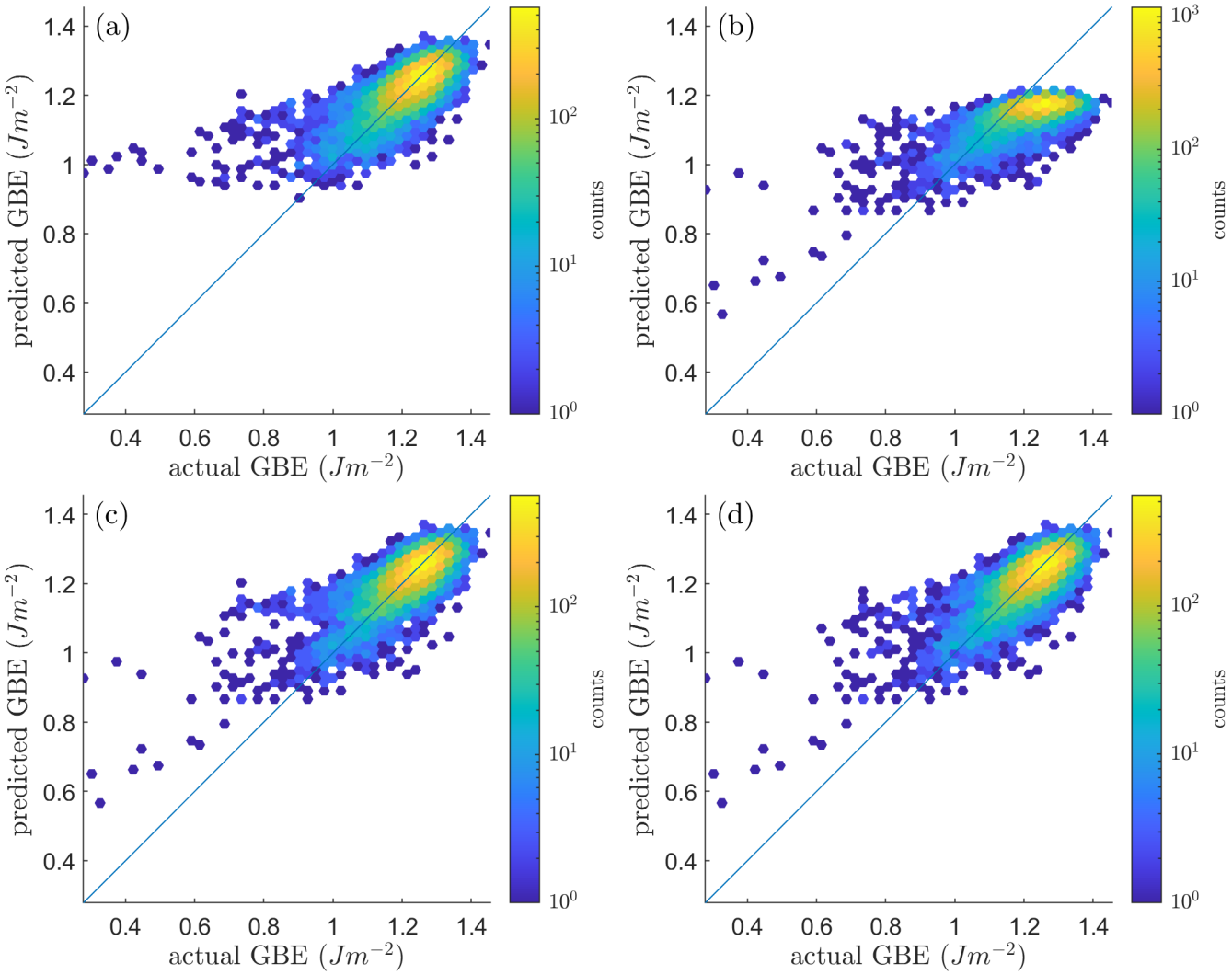

Figure S7: (a) Hexagonally binned parity plot of the standard GPR model. (b) All prediction GBs based on the model using only training GBs with a GBE less than 1.2 J m^{-2}. (c) Combined disjoint model as explained in the text. (d) Hexagonally binned parity plots of the final GPR mixing model. Points in (c) are produced by splitting the prediction data into less than and greater than 1.1 J m^{-2}. A sigmoid mixing function (Figure S8) is then applied where the predicted GBEs shown in (c) determines the mixing fraction (f) to produce a weighted average of models (a) and (b). A large Fe simulation database [8] using 46 883 training datapoints and 11 721 validation datapoints in an 80%/20% split. The GPR mixture model decreases error for low GBE and changes overall RMSE and mean absolute error from 0.057 932 J m^{-2} and 0.039 857 J m^{-2} in the original model (shown in (a)) to 0.055 035 J m^{-2} and 0.039 185 J m^{-2} (shown in (d)), respectively.

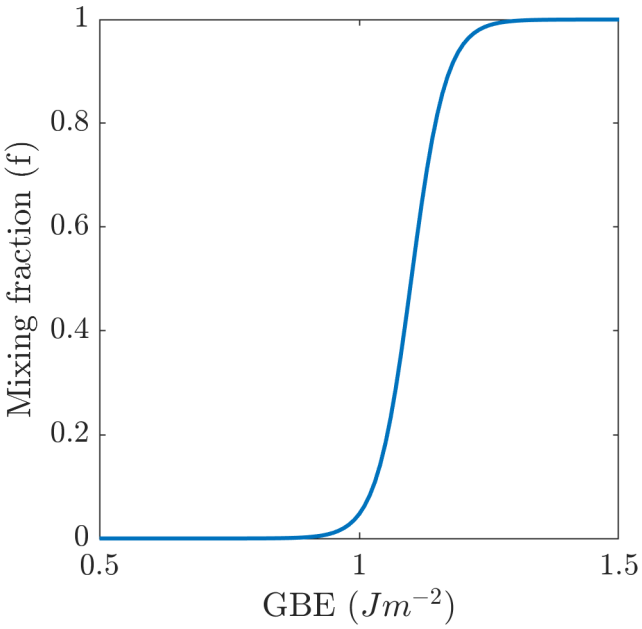

Figure S8: Sigmoid mixing function used in the GPR mixing model with $m = 30$ and $b = 1.1\,\mathrm{J\,m^{-2}}$ (Eq. (S2)).

S7. Olmsted Interpolation

As illustrated in Figure S9, leave-one-out cross validation interpolation results for 0 K molecular statics low-noise Ni simulations using the GPR method are similar to Laplacian kernel regression results reported in Figure 6a of Chesser et al. [5] (reproduced on the right of Figure S9 for convenience).

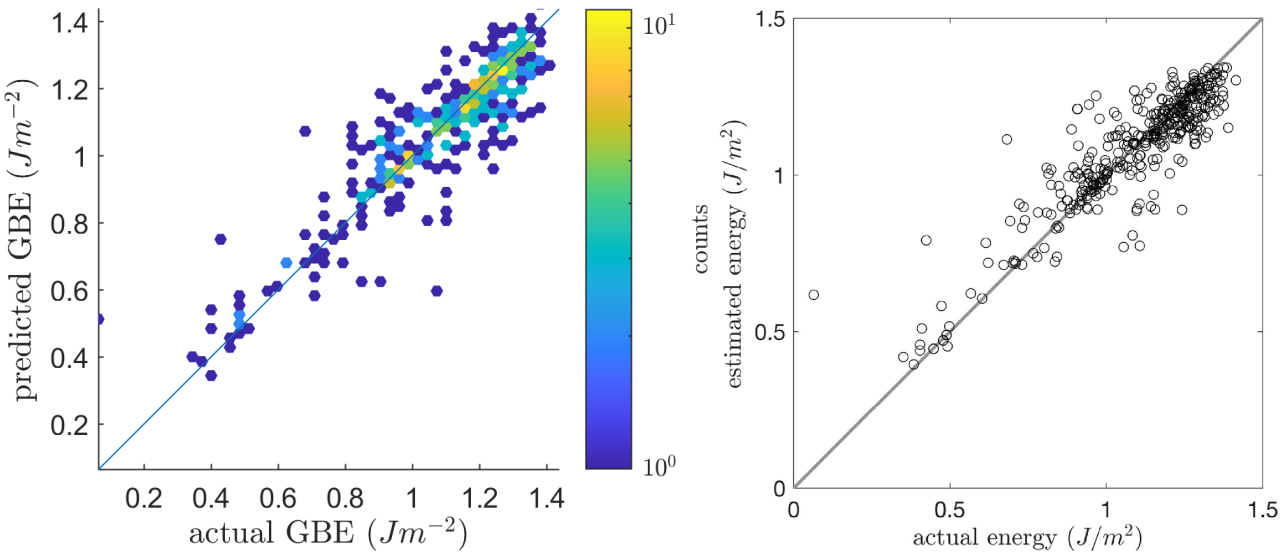

Figure S9: (left) Hexagonally binned parity plot for Ni simulation grain boundary energy (GBE) interpolation using LOOCV. (right) Parity plot for leave-one-out cross validation (LOOCV) interpolation results reproduced from Figure 6a of Chesser et al. [5] under CC-BY Creative Commons license.

Glossary

GB grain boundary 2–7, 9–11

GBE grain boundary energy 2, 5, 6, 8–11, 13

GBO grain boundary octonion 2–8, 10

GPR Gaussian process regression 2, 5–12

LOOCV leave-one-out cross validation 13

NN nearest neighbor 2, 5, 6

RMSE root mean square error 3, 8, 11

SEO symmetrically equivalent octonion 3

VFZ Voronoi fundamental zone 2–4, 6–8, 10

VFZ-GBO Voronoi fundamental zone grain boundary octonion 2–4, 6–9

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.