

1      GrainNN: A neighbor-aware LSTM framework for predicting  
2      microstructure evolution during polycrystalline grain formation

3      Yigong Qin\*    Stephen DeWitt†    Balasubramanian Radhakrishnan‡    and George Biros §

4      **Abstract**

5      High fidelity simulations of grain formation in alloys are an indispensable tool for process-to-  
6      mechanical-properties characterization. However, such simulations require fine spatial and temporal  
7      discretizations that can be computationally expensive. This cost is an obstacle to parametric stud-  
8      ies and ensemble runs and ultimately makes downstream tasks like optimal control and uncertainty  
9      quantification challenging. To enable such simulations, we introduce GrainNN, an efficient and accu-  
10     rate reduced-order model for epitaxial grain growth in additive manufacturing conditions. GrainNN  
11     is a sequence-to-sequence long-short-term-memory (LSTM) deep neural network that evolves the  
12     dynamics of manually crafted features. Its innovations are (1) an attention mechanism with grain-  
13     microstructure-specific transformer architecture; and (2) an overlapping combination of several clones  
14     of the network to generalize to grain configurations that are different from those used for training.  
15     This design enables GrainNN to predict grain formation for unseen physical parameters, grain number,  
16     domain size and geometry. Furthermore, GrainNN not only reconstructs the quantities of interest  
17     but also can be pointwise accurate. In our numerical experiments, we use a polycrystalline phase  
18     field method to both generate the training data and assess GrainNN. For multiparametric, ensemble  
19     simulations with many grains, GrainNN can be orders of magnitude faster than phase field simula-  
20     tions, while delivering 5%–15% pointwise error. This speedup includes the cost of the phase field  
21     simulations for generating training data.

22     **1 INTRODUCTION**

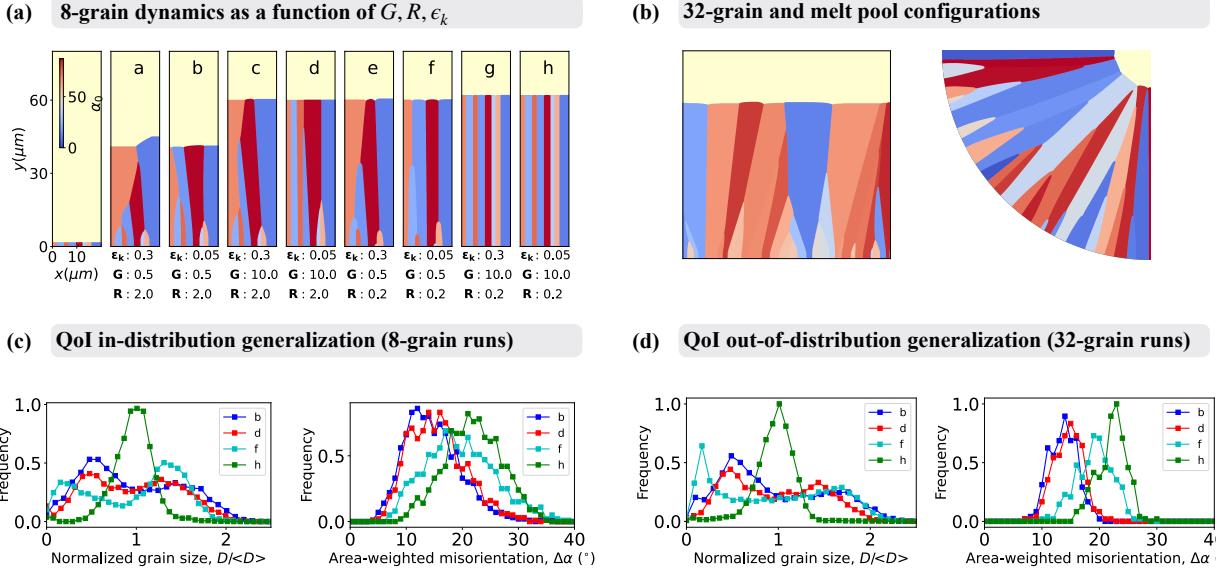
23     Predicting grain formation during alloy solidification is of great importance in additive manufacturing  
24     (AM) and other processes as the microstructure controls mechanical properties of the built com-  
25     ponents [1–4]. High fidelity full-order model simulations (FOMS) like phase field partial differential  
26     equations (PDEs) [4–8], kinetic Monte Carlo [9], cellular automata [10, 11] and other methods are  
27     used to guide the development of processing-microstructure linkages in alloy manufacturing [12].  
28     Tasks like process optimization and parametric uncertainty quantification (UQ) require ensemble  
29     FOMS [8, 13–15] to statistically characterize quantities of interest (QoIs) like grain size, aspect  
30     ratio, and misorientation [16–18]. But owing to the multiple physics and spatiotemporal scales,  
31     FOMS remain particularly expensive despite tremendous advances in algorithms, software, and hard-  
32     ware [19–22]. The high cost of a single simulation compounded with the need for a large number of  
33     simulations has restricted the adoption of FOMS for the formation of solidification microstructure in  
34     production workflows. An alternative to FOMS is reduced-order models (ROMs). We are particularly

\*Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712, USA; ygqin@utexas.edu

†Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA;  
dewittsj@ornl.gov

‡Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA;  
radhakrishnb@ornl.gov

§Oden Institute, The University of Texas at Austin, Austin, TX 78712, USA; gbiros@acm.org



**Fig. 1 Overall problem description.** We designed *GrainNN* to enable ensemble simulations that can predict statistics of microstructure morphology. **(a):** To reduce training data generation cost, we use FOMS of a small, eight-grain rectangular geometry, parameterized by the temperature gradient  $G$ , pulling velocity  $R$ , kinetic anisotropy  $\epsilon_k$ , and the random multidimensional variable  $\xi$  for width and orientation of the partly melted initial grains in the metal substrate. **(b):** Our target configurations have an arbitrary number of grains solidifying in a possibly non-rectangular geometry. Our goal is to avoid FOMS of such configurations. **(c):** Ensemble runs over  $\xi$  result in QoI histograms shown here for four of the configurations in (a). Each curve corresponds to specific  $G$ ,  $R$ ,  $\epsilon_k$  values and is obtained using 1000 FOMS for different  $\xi$ . **(d):** Here we show QoI histograms for the 32-grain rectangular configuration in subfigure (b) for different solidification conditions. Each curve is computed using 250 *GrainNN* inference runs that sample  $\xi$ . Producing these QoI histograms for 200 different  $G$ ,  $R$ ,  $\epsilon_k$  values using *GrainNN* is over 1000X faster than using our grain-scale polycrystalline phase field FOMS.

interested in *data-driven* ROMs [23], which are calibrated or “trained” using FOMS or experimental data. For a range of applications, such ROMs have been successful in addressing challenges in uncertainty quantification and optimization [24–27]. However, AM conditions involve highly nonlinear, non-homogeneous, non-equilibrium, stochastic, and evolving-interface processes, and as a result classical ROM methods [28] have not been as successful.

Beyond classical ROMs, methods based on machine learning have received increasing attention in recent years. Examples include microstructure classification [29, 30], recognition [31], reconstruction [32, 33], optimization [34], and structure-to-material-properties prediction [35, 36]. Convolutional neural networks have been widely used to extract microstructure features and reduce the spatial dimension of imaging data [34, 37, 38]. Generative adversarial networks [39] are capable of learning microstructure statistics, and can approximate structure-to-material properties forward and inverse maps [40, 41]. All these works have focused on static microstructure images. There is less work on ROMs for microstructure evolution. Existing works consider a representative volume element and use long short-term memory (LSTM) [42] networks to capture evolution statistics. Examples include ROMs for spinoidal decomposition of binary mixtures [43, 44], single dendrite growth [44], and grain coarsening [45]. Yet a *pointwise* accurate ROM for grain formation in AM is lacking.

In this paper, we introduce such a pointwise accurate ROM; it is an LSTM-based ROM, which we term “*GrainNN*”. We focus on 2D epitaxial growth without nucleation as seen in Fig. 1. In many additively manufactured alloys, the solidification microstructure occurs by epitaxial growth of the grains from the base plate [46, 47]. The parameters that are significant in controlling the morphology of grains during epitaxial growth [8, 46, 48] include temperature gradient  $G$ , pulling velocity  $R$ , and kinetic anisotropy  $\epsilon_k$ . Then, the input to *GrainNN* is  $p = \{G, R, \epsilon_k\}$ ; and a random

initial grain state  $\xi = \{\alpha_{i0}, w_{i0}\}_{i=1}^{N_G}$  (where  $\alpha_{i0}$ ,  $w_{i0}$  are the  $i$ <sub>th</sub> grain orientation and width, and  $N_G$  is the number of grains). The output is the spatiotemporal history of the microstructure evolution. GrainNN has the following characteristics: (a) It captures the evolution dynamics of manually-defined grain growth features using a *novel physics-interpretable* network architecture. (b) It can predict pointwise microstructure and QoI distribution for unseen  $p$ , and  $\xi$  (as in Fig. 1a). We refer to this capability as *in-distribution generalization*, because although these values are unseen, they come from the same probability distribution of the values we selected for training. (c) It can generalize with respect to the time horizon, the number of grains, the probability distribution of the random variable  $\xi$ , and domain geometry (Fig. 1b). We refer to this capability as *out-of-distribution generalization*, because we never sample such configurations during training.

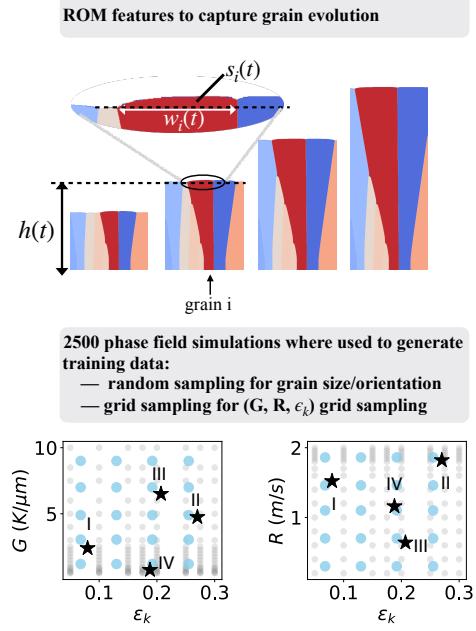
GrainNN uses two LSTM sub-networks to model the dynamic evolution of the defined features: a one-to-sequence LSTM that predicts the first sequence from initial conditions; and a sequence-to-sequence LSTM [49] that evolves the microstructure. The one-to-sequence LSTM circumvents the need to run FOMS to compute the first sequence required in the sequence-to-sequence LSTM. Then we switch to a sequence-to-sequence LSTM because we found it outperforms one-to-sequence networks. To capture spatial correlations between grains of different orientations, a natural choice would be to add convolutional layers to LSTM architecture [50], which are quite common in spatiotemporal predictions [44, 51]. However convolutional operations are spatially invariant and local. During grain formation, grains can be *eliminated* and the interactions are in fact *non-local* and also *vary in time* (Fig. 1b). In GrainNN, we introduce a novel *self-attention* mechanism inspired by transformer architectures used in text [52, 53] and vision [54, 55]. We embed this self-attention mechanism to the LSTM to create, what we call, a neighbor-aware LSTM (NA-LSTM), which allows GrainNN to find and couple the time-dependent neighbors of every grain. We compare the performance of NA-LSTM with a standard LSTM in the supplementary.

A ROM is efficient only if the FOMS cost to generate training data is much smaller than the FOMS cost of just doing the target computation (say UQ). To reduce this cost we use a "base" eight-grain ROM in a rectangular geometry, as shown in Fig. 1a. The solid substrate comprises eight grains of random width and orientation at the bottom with no undercooling. A heat source with constant  $G$  and  $R$  is applied in the vertical direction and is stopped when the interface velocity assumes a steady state equal to the pulling velocity.

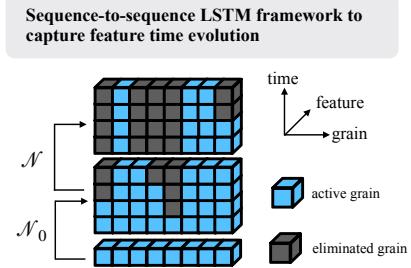
To use the base ROM for inference on configurations with more than eight grains and/or non-rectangular geometries, we couple multiple base ROMs and use rectangular-to-curvilinear domain maps. In this way, we can simulate entire spot-weld melt pools without additional training. This is inspired by a recent study by our group [56] in which we compared dendrite-resolving FOMS in a full melt pool with phase field simulations in a narrow rectangular domain. QoIs such as interface undercooling and moving speed of the narrow-domain configuration showed excellent agreement with the full-melt-pool configuration, which suggests our base model-to-melt pool geometry generalization.

We empirically demonstrate that GrainNN captures grain boundary evolution and is capable of accurate pointwise predictions. Besides the exact microstructure, we also like to evaluate GrainNN on its ability to predict QoI probability distributions under various solidification conditions (Fig. 1c and Fig. 1d). The first QoI is the normalized grain size distribution  $\{D_i/\langle D \rangle\}$  [15], where the size of grain  $i$  is defined by its area-equivalent diameter  $D_i = (4A_i/\pi)^{1/2}$ ,  $\langle D \rangle$  denotes the average grain size, and  $A_i$  the  $i$ <sub>th</sub> grain area. The second QoI is the area-weighted misorientation  $\Delta\alpha$  that quantifies the degree of grain competition during epitaxial growth.  $\Delta\alpha = \sum_i A_i |\alpha_{i0} - \alpha_G| / \sum_i A_i$  and  $|\alpha_{i0} - \alpha_G|$  is the angle between crystal orientation  $\alpha_{i0}$  and prescribed temperature gradient direction  $\alpha_G$ . Smaller  $\Delta\alpha$  means that a larger proportion of grains is aligned with the pulling direction. In Fig. 1c, 1000 phase field realizations are conducted to ensure statistical convergence of the QoI histograms. GrainNN achieves orders-of-magnitude speedup over FOMS when generating ensemble microstructures for unseen geometries/configurations, even after we account for the cost of FOMS to generate the training data.

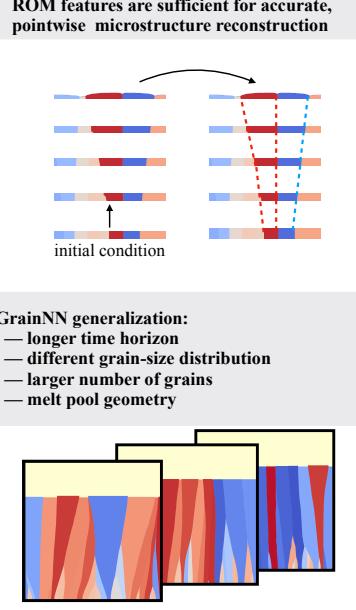
**(a) Data generation**



**(b) Learning feature evolution**



**(c) Prediction**



**Fig. 2 Training, learning mechanisms, and predictions of GrainNN.** **(a) Top:** We use  $2N_G + 1$  features per (coarse) time step to describe the polycrystalline growth of  $N_G$  grains:  $h(t)$  is the height of an across-grain interface (the dashed black line). For each grain  $i$ ,  $w_i(t)$  is its width at  $h(t)$ ; and  $s_i(t)$  is the excess area (the area above the across-grain interface). **(a) Bottom:** We run 2500 (2400 for training, 100 for validation) grain-scale phase field PDE simulations to train GrainNN, each run with a different  $p = \{G, R, \epsilon_k\}$  (faded-gray-dots grid) and a randomly-sampled  $\xi$  per  $p$ . The blue dots represent 100  $p$  values used to test GrainNN pointwise accuracy for different  $p$ . The black stars I, II, III, IV are four  $p$  values that we used to test GrainNN pointwise accuracy for different  $\xi$ ; for each  $p$  we sample 100  $\xi$ . **(b) Top:** GrainNN comprises two NA-LSTM networks:  $\mathcal{N}_0$  initiates the feature evolution and  $\mathcal{N}$  applies a sequence-to-sequence LSTM. Each column represents a different grain and each row a different time. Blue grains are “active”, meaning their  $w(t) > 0$ . Gray grains are “eliminated”, with  $w(t) = 0$ . **(b) Bottom:** The NA-LSTM has a transformer component that correctly captures the time-dependent left and right neighbors for each grain  $i$ . **(c) Top:** The full microstructure can be reconstructed with the defined grain features by interpolation. **(c) Bottom:** GrainNN is able to predict unseen configurations including generalization in time horizon, number of grains, and domain size and geometry.

## 2 RESULTS

We would like to emphasize that GrainNN does not depend on the high-fidelity simulation method used to generate grain-level microstructure evolution. Here, we use a grain-scale, multiple-phase field method as our FOM [57]. Grain-scale phase field methods have been extensively applied in grain coarsening [13–15] and grain growth with a heat source [7, 8, 57, 58]. We generate the training dataset by sampling  $G$  in  $(0.5, 10) K/\mu m$  [4, 7, 8],  $R$  in  $(0.2, 2) m/s$  [8, 58], and kinetic anisotropy  $\epsilon_k$  in  $(0.05, 0.3)$ . The sampling of  $\epsilon_k$  represents uncertainty in material properties [4, 8, 59]. In Fig. 1a, configurations *a–h* show the microstructure at the end of the simulation with different combinations of  $G$ ,  $R$ , and  $\epsilon_k$ . Fig. 1b is our target configuration, the goal is to accurately predict microstructure of systems of melt pool size and geometry using the base model trained by the configuration in Fig. 1a. We present results for stainless steel 316L. As we demonstrate in the supplementary, GrainNN generalizes to other alloys.

119        **2.1 GrainNN structure and training**

120        Instead of using automatic feature extraction methods, we use manually defined, explainable, time-  
121        varying features defined at the solid-liquid interface (SLI). As shown in Fig. 2a, we define  $h(t)$  to be  
122        the height of an across-grain interface, which we define as the height of the *lowest SLI point*. For  
123        each grain  $i$ , we define its grain width  $w_i(t)$  at  $h(t)$  and the excess area  $s_i(t)$  above  $h(t)$ . In short,  
124        we track  $2N_G + 1$  features:  $\{w_i(t), s_i(t)\}_{i=1}^{N_G}$  and  $h(t)$ , where  $N_G$  is the number of grains. With the  
125        time history of these features, we can reconstruct the grain microstructure evolution and compute  
126        the quantities of interest. The time step size in GrainNN is much coarser than the one used in FOM;  
127        GrainNN typically takes less than thirty time snapshots.

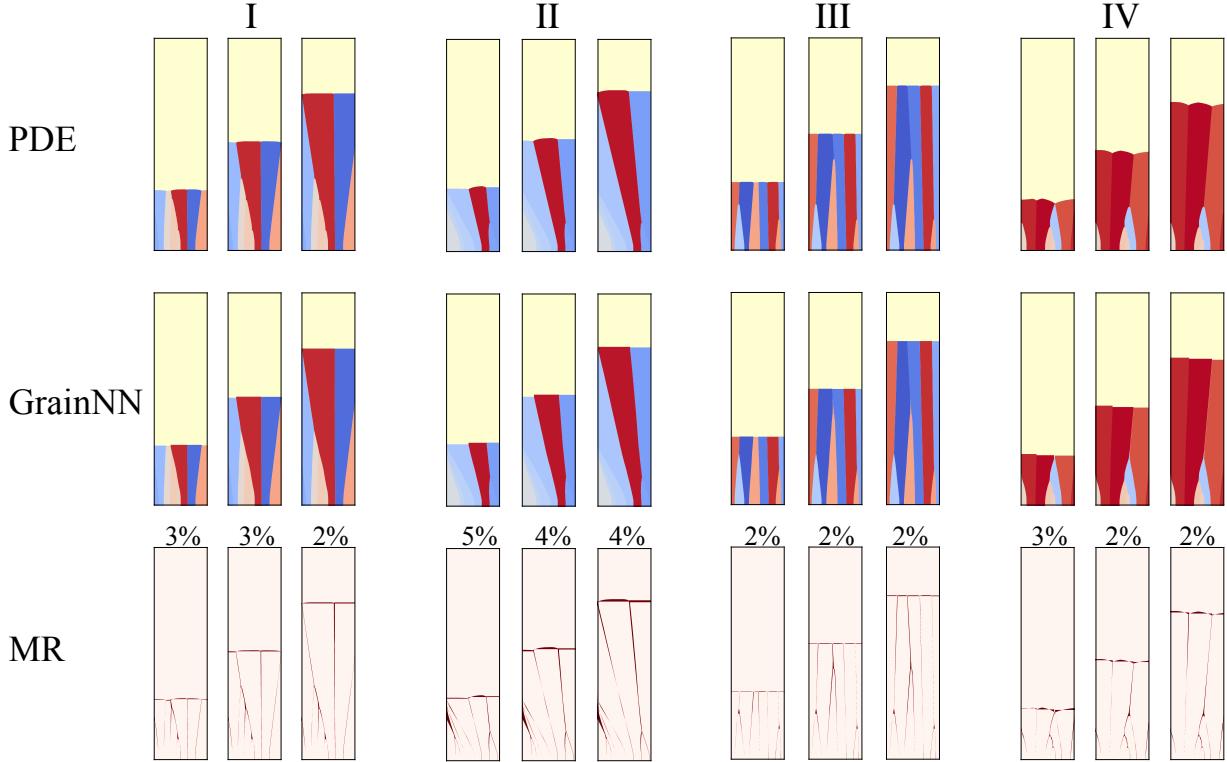
128        The structure of GrainNN is summarized in Fig. 2b. It consists of  $\mathcal{N}_0$ , the one-to-sequence  
129        network that takes the initial conditions of the features and produces a sequence of time snapshots,  
130        and  $\mathcal{N}$  that maps sequences to sequences. Each cube in Fig. 2b represents the features of a grain  $i$   
131        at time  $t$ . Columns represent different grains, and each row is a discrete GrainNN time step. Grains  
132        with  $w_i(t) > 0$  are “active”, otherwise they are “eliminated”. The single-layer GrainNN architecture is  
133        given in *Methods*; the full multilayer GrainNN is given in the Supplementary. We train 243 GrainNN  
134        models with different network architecture hyperparameters. (The total training time was 11 hours  
135        on four AMD EPYC 7763 64-Core processors.) We chose the top four models—in terms of validation  
136        accuracy, typically  $10^{-3}$  relative reduction in loss function—and we use them for ensemble inference  
137        by feature averaging across their GrainNN predictions. Each GrainNN is parameterized by roughly  
138        300K network weights, 200K for  $\mathcal{N}$  and 100K for  $\mathcal{N}_0$ .

139        We ran 2500 phase field simulations (2400 for training and 100 for validation) to generate  
140        microstructures from which we extract grain features to train GrainNN. Each simulation uses a  
141        different  $p$  and  $\xi$ . We used a 3D grid sampling in  $p$  space with 20  $G$ , 15  $R$ , and 8  $\epsilon_k$  values. The  
142        regular-grid  $p$  values for training are shown with faded gray dots in Fig. 2a. Our sampling is denser in  
143         $p$  regions with stronger grain competition. The total FOM cost for generating training data is about  
144        40 hours on one NVIDIA Tesla 100 GPU card (Supplementary Table 2). We also ran additional FOM  
145        simulations to test the accuracy of GrainNN.  $p$  values of 500 test runs are shown in Fig. 2a.

146        Next, we present the test errors for both pointwise and QoI predictions. Pointwise errors are  
147        measured using the misclassification rate (MR). MR is defined for some time  $t$  as the number of grid  
148        points classified to a wrong grain divided by the area of a rectangle defined by the maximum SLI  
149        height at time  $t$ . MR is zero only when all grains are exactly reconstructed. We also report per-grain  
150        accuracy by calculating Sørensen–Dice coefficient (DICE) [60] for each individual grain. Given the  
151        real shape of a grain and the predicted shape, DICE measures the percentage of their overlapped  
152        area. Perfect predictions correspond to DICE being equal to one. For the QoI ensemble predictions  
153        we define two error metrics. The first metric is the relative error of the expected value. The second  
154        metric is a distributional error, here we use the two-sample Kolmogorov–Smirnov (KS) test, which  
155        quantifies the maximum discrepancy between ground-truth and predicted QoI distributions. The  
156        smaller the KS value the closer the two distributions are.

157        **2.2 In-distribution generalization**

158        As discussed, in-distribution generalization refers to eight-grain predictions for unseen during training  
159         $p$  and with  $\xi$  drawn from the same probability distribution used for training. Fig. 3 shows the error  
160        of microstructure prediction for the four different values of  $p$ . The first row shows the phase field  
161        results (spatial resolution is  $625 \times 2500$ ). Plots from left to right correspond to three different time  
162        snapshots for  $p_I$ – $p_{IV}$ . The second row shows the corresponding GrainNN predictions starting with  
163        the same initial condition as the first row. In the third row, we show the pointwise error where  
164        misclassified grid points are shown in dark red. The microstructures predicted by GrainNN are highly  
165        accurate representations of the microstructures obtained by phase field simulations. We observe MR  
166        is less than 5% for all time snapshots. One drawback of the GrainNN prediction is the loss of surface  
167        shape details. For example, in the  $p_{II}$  case the curvatures of the grain top surfaces can not be  
168        predicted by GrainNN because we only track excess area and we use rectangles to approximate the  
169        part of grain which is higher than the across-grain interface. In addition to the simulations shown in

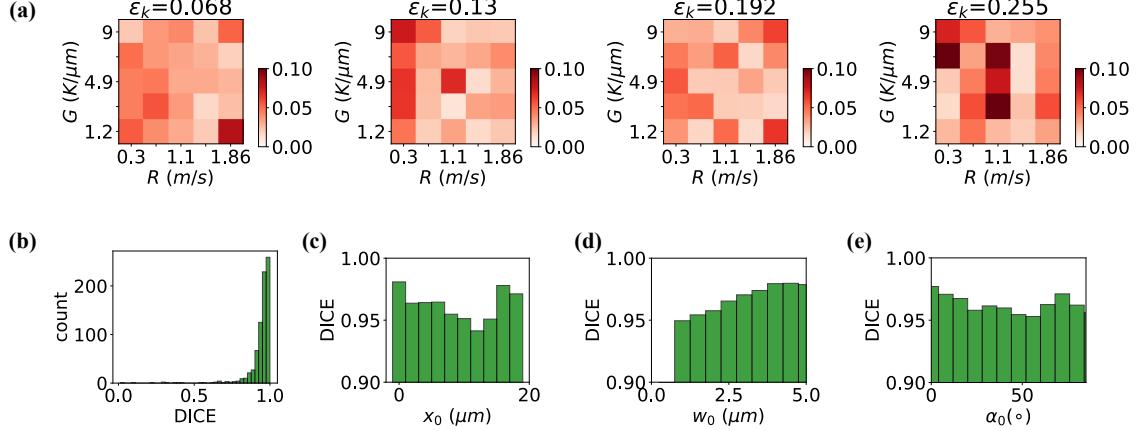


**Fig. 3 Error of morphology prediction compared to phase field simulations for testing parameters  $p_I$ – $p_{IV}$ .** The first row is the PDE simulations at three different time snapshots. The second row is the corresponding GrainNN predictions. The last row shows the pixel-wise discrepancy between the two images above. The four testing parameters are the black stars shown in Fig. 2.  $p_I$ :  $G = 2.4K/\mu\text{m}$ ,  $R = 1.52\text{m}/\text{s}$ ,  $\epsilon_k = 0.08$ ;  $p_{II}$ :  $G = 4.75K/\mu\text{m}$ ,  $R = 1.82\text{m}/\text{s}$ ,  $\epsilon_k = 0.27$ ;  $p_{III}$ :  $G = 6.5K/\mu\text{m}$ ,  $R = 0.64\text{m}/\text{s}$ ,  $\epsilon_k = 0.207$ ;  $p_{IV}$ :  $G = 0.75K/\mu\text{m}$ ,  $R = 1.16\text{m}/\text{s}$ ,  $\epsilon_k = 0.188$ .

170 Fig. 3, we measure MR error by sampling  $\xi$  values for each of the four  $p$  values. Their MR average  
 171 and standard deviation are  $3.5 \pm 1.6\%$ ,  $4.2 \pm 3.6\%$ ,  $4.1 \pm 2.0\%$ , and  $4.5 \pm 3.1\%$  respectively.

172 We further test GrainNN using 100 different  $p$  values (the blue dots in Fig. 2a). The average  
 173 misclassification rate for the 100 testing points is 3.7% with 77 of the cases under 5%. The maximum  
 174 MR is 10%. We don't observe a correlation between MR and the sampling location in the  $p$  space  
 175 although the maximum error is observed for  $\epsilon_k = 0.255$ . Fig. 4b shows DICE for all 800 grains at the  
 176 final time of each simulation. Among them, 681 grains (85%) have DICE larger than 0.9 and the  
 177 average DICE is about 0.95. In Fig. 4c-e, we investigate DICE dependency on base grain properties.  
 178 We find the DICE coefficient is higher for grains with smaller boundary-to-area ratios. Most grains  
 179 with larger initial width  $w_0$  end up having a larger area at the end. For grains near the left and right  
 180 boundaries of the domain, i.e.,  $x_0=0$  or  $20 \mu\text{m}$ , part of their grain boundaries is already fixed by the  
 181 domain and thus leads to higher accuracy.

182 We report QoI errors for the eight  $p$  values from Fig. 1a. The FOM QoIs were sampled using 1000  
 183  $\xi$  phase field realizations for each  $p$ . The ROM QoIs were sampled using 1000 GrainNN inferences by  
 184 resampling  $\xi$ . In Table 1,  $\kappa_D$  and  $\kappa_{\Delta\alpha}$  are KS test values for grain size distribution and area-weighted  
 185 misorientation distribution, respectively. As mentioned before, a smaller KS is better. For all cases  
 186 KS is less than 0.075.  $\epsilon_{<D>}$  and  $\epsilon_{<\Delta\alpha>}$  are the relative error of the QoI expectations. The largest  
 187 errors are about 7% for the (f) and (g) cases, but otherwise they are less than 4% for both QoIs.  
 188 One question we had is what is the number of FOM simulations to compute the QoIs to the same  
 189 accuracy as the ROM —treating 1000-FOM QoI values as the ground truth. To do so, we randomly  
 190 remove data from the 1000 FOM simulations until one of the error metrics exceeds the ROM error.



**Fig. 4 Pointwise errors for 100 testing  $p$ .** (a) The 100  $p$  values are the combinations of five  $G$ , five  $R$ , and four  $\epsilon_k$  values. For each one we show misclassification rates at the final simulation time. Each point in the error heatmap corresponds to one  $\xi$  per  $p$ . (b) Histogram of Sørensen–Dice coefficient (DICE) for each grain.  $DICE=1$  means a perfect reconstruction of the grain shape. (c) DICE as a function of initial grain location,  $x_0$  is the x-coordinate of the center of a grain. (d) DICE as a function of initial grain width  $w_0$ . (e) DICE as a function of grain orientation  $\alpha_0$ .

**Table 1** Prediction accuracy of quantities of interest (QoI) distributions for eight  $p$  values. For each  $p$  we sample 1000  $\xi$  for phase field simulations and another 1000  $\xi$  for GrainNN inferences. Two-sample Kolmogorov–Smirnov (KS) tests are used to compute the distribution mismatches for each  $p$ .  $\kappa_D$  and  $\kappa_{\Delta\alpha}$  are the KS statistics of normalized grain size distribution and area-weighted misorientation distribution.  $\epsilon_{<D>}$  and  $\epsilon_{<\Delta\alpha>}$  are the relative error of expectations.

Error metrics		$p$ values (see Fig. 1)							
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
KS statistics	$\kappa_D$	0.014	0.024	0.024	0.019	0.069	0.062	0.047	0.026
	$\kappa_{\Delta\alpha}$	0.045	0.062	0.038	0.050	0.026	0.033	0.071	0.035
Error of expectation	$\epsilon_{<D>}$	0.010	0.013	0.012	0.008	0.074	0.072	0.016	0.007
	$\epsilon_{<\Delta\alpha>}$	0.036	0.017	0.001	0.024	0.005	0.015	0.027	0.009
#FOMS to reach ROM error		300	300	1000	300	900	600	200	300

191 This is reported in the last row. The total number of simulations required for the eight cases is 3900,  
 192 which is actually larger than the number of the FOM simulations used for training and this is only  
 193 for eight  $p$  values. Since GrainNN inference time is negligible compared to the data-generation time,  
 194 this experiment shows training a ROM is useful even for a small number of parametric studies. Of  
 195 course what more valuable is GrainNN's applicability to problems with a large number of grains and  
 196 non-planar configurations.

### 197 2.3 Out-of-distribution generalization

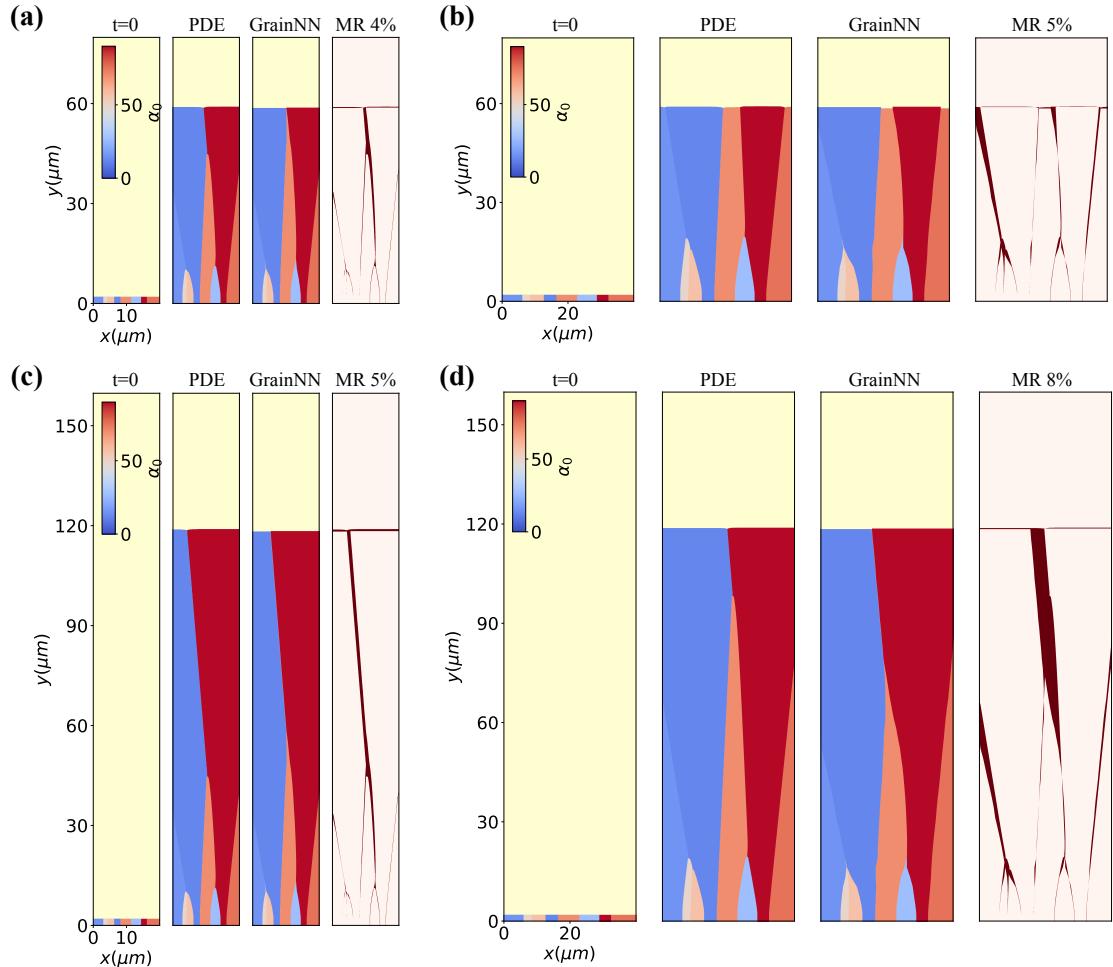
198 We trained GrainNN using ground truth FOM data with a  $L_x = 20\mu m$ -wide rectangular domain,  
 199  $N_G = 8$  with mean substrate grain size  $w_\mu = 2.5\mu m$ , and time horizon when the isotherm reaches  
 200  $62\mu m$ . Here we test GrainNN's ability to predict grain formation and the related Qols for longer  
 201 time horizons, larger  $w_\mu$  (larger  $L_x$ ), larger  $N_G$ , and a non-rectangular geometry. We remark that  
 202 these generalizations do not require retraining but they're not automatic: they require either small  
 203 modifications, e.g., scaling in the base GrainNN or additional algorithmic innovation, e.g., combining  
 204 multiple GrainNNs. We explain these innovations below.

205 **Time horizon:** This is the simplest generalization as we simply run GrainNN for more time  
 206 steps. Since we've trained GrainNN long enough for the SLI to reach a "steady-state" velocity, grain  
 207 formation assumes a steady-state pattern. We speculate that the last couple of time frames can  
 208 be used as a "template" for  $t$  greater than training time horizon  $t_H$  without any modifications to  
 209 GrainNN. Fig. 5 shows such an example of time "extrapolation". Fig. 5a shows one testing case at  
 210 time  $t_H$ , and Fig. 5c extrapolates it to  $2t_H$ . We apply this extrapolation to  $2t_H$  for 100 different  $p$ .  
 211 Their average MR increases from 3.7% for  $t_H$  to 5.4% for  $2t_H$ .

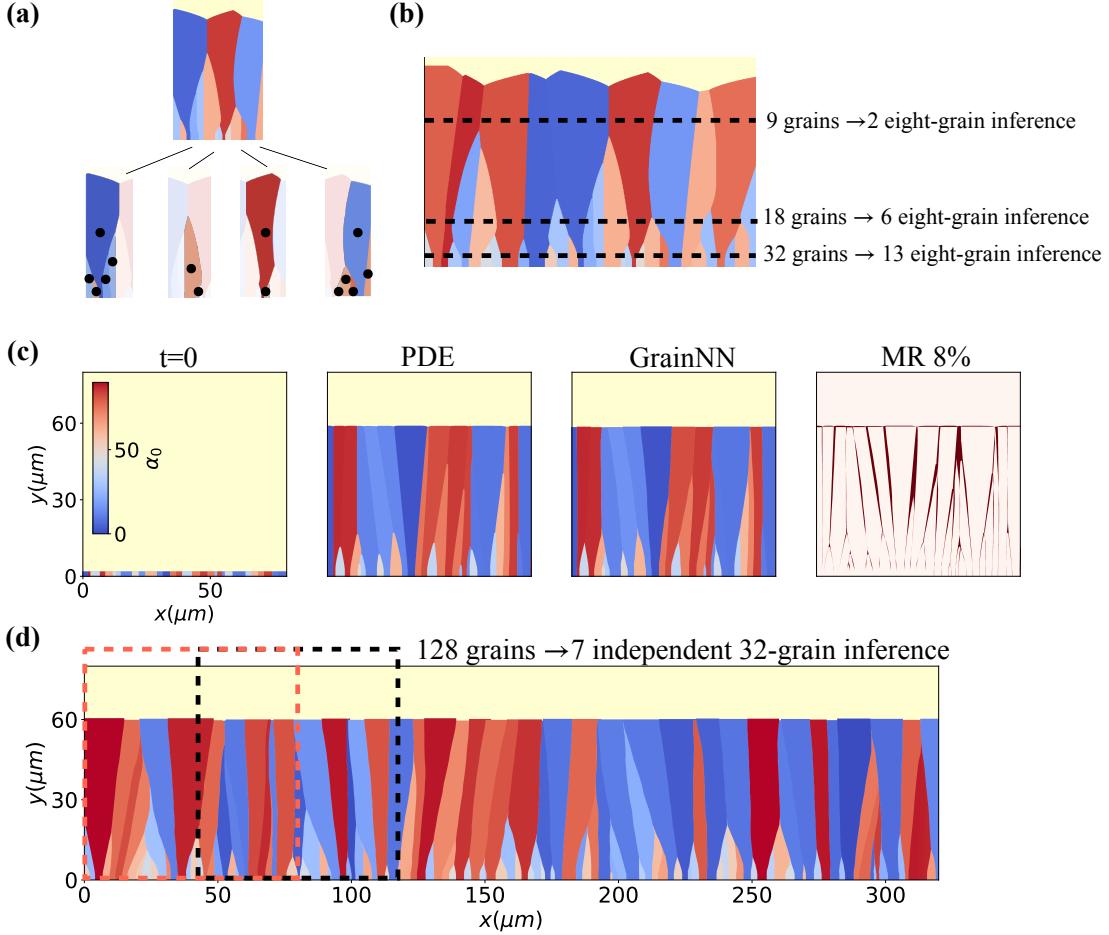
212 **Domain size  $L_x$ :** Different  $L_x$  with the same number of grains  $N_G$  is equivalent to different initial  
 213 mean grain width ( $\xi$  distribution). As we explain in *Methods*, we non-dimensionalize both phase  
 214 field PDE and GrainNN. Then, changing  $L_x$  means changing parameters that appear nonlinearly in  
 215 the PDE, and this would require training GrainNN for unseen  $L_x$ . Instead of additional training, we  
 216 assume that the "vertical" grain-to-grain interface slope is not as sensitive to the grain width and  
 217 mostly depends on alloy, grain orientation, and temperature. The assumption of width independence  
 218 allows us to introduce a scaling in the grain width GrainNN updates. Specifically, we introduce a  
 219 domain scaling factor  $c_l = L_x/L_x^{\text{training}}$  that adjusts GrainNN output and makes it invariant to  $L_x$   
 220 (see Eq. (4.6a) in *Methods*). We also assume that the excess-area-to-width ratio  $s_i/w_i$  is invariant  
 221 and use it to scale  $s_i$ . Fig. 5b is an example with  $c_l = 2$  ( $L_x = 40\mu m$ ). We compare with 100 phase  
 222 field simulations with the testing  $p$ ; the average MR increases from 3.7% for  $c_l = 1$  to 7.6% for  
 223  $c_l = 2$ . In Fig. 5d, we combine longer time horizons and wider domain generalization.

224 **Number of grains:** To allow the use of GrainNN for an arbitrary number of grains we use a  
 225 domain decomposition approach: first we group the grains in *overlapping* eight-grain subgroups so  
 226 that each subgroup is time-advanced independently using the pretrained GrainNN. Each subgroup  
 227 has a six-grain overlap with its left subgroup and a six-grain overlap with its right neighborhood.  
 228 Recall that one GrainNN iteration creates a sequence of time snapshots. After one GrainNN iteration,  
 229 we only keep the two middle grains from each subgroup. Without grain elimination, this allows us to  
 230 continue the microstructure evolution without changing the grouping. However, in the presence of  
 231 grain elimination, we need to redefine the subgroups. Another technical issue is that we need to ensure  
 232 that all grain widths add up to the domain size. So after the subgroup step, we need to renormalize  
 233 widths. So in summary the algorithm has the following steps: (1) At each GrainNN iteration,  
 234 define subgroups with six-grain left/right overlap; (2) GrainNN advances subgroups independently;  
 235 (3) Update global microstructure using the two middle grains from each subgroup; (4) Renormalize  
 236 widths to ensure correct domain width; (5) Repeat.

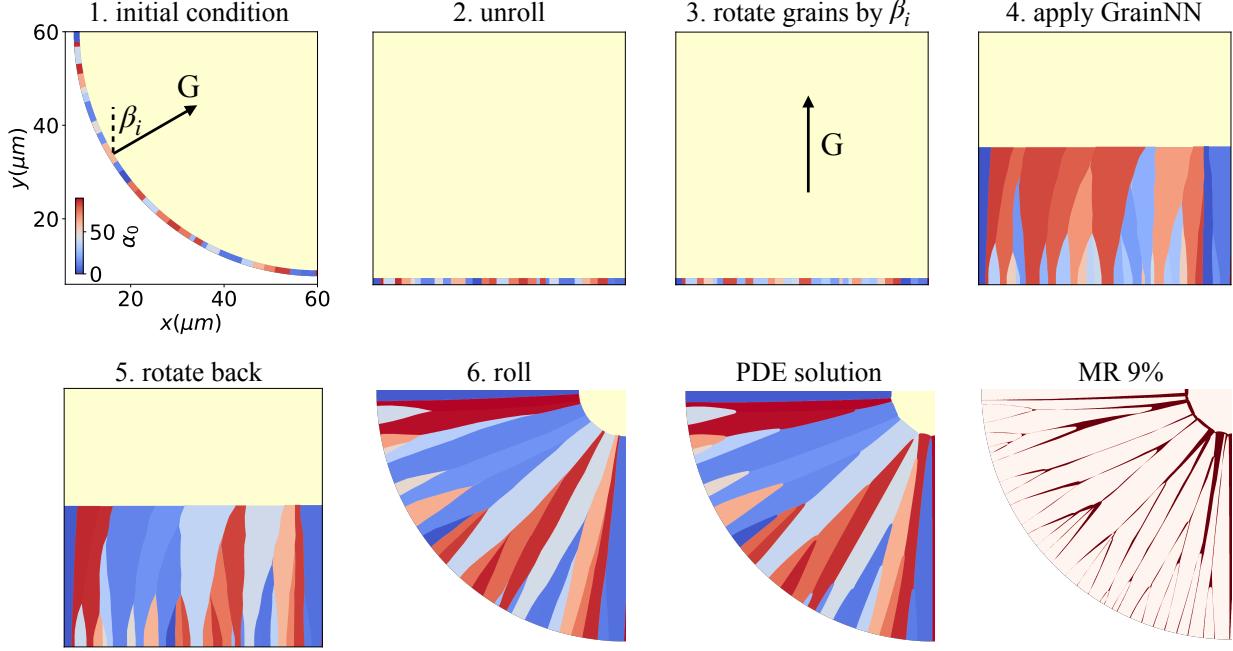
237 Fig. 6a shows an example of a 14-grain system partitioned into four eight-grain subgroups. Fig. 6b  
 238 shows the scheme being applied to a 32-grain original configuration. Some of the grains may be  
 239 eliminated during a step, so we repartition the entire system based on the number of active grains at  
 240 the SLI. As shown in Fig. 6b, the initial number of active gains is 32 and the number of partitions is  
 241 13; at the next GrainNN iteration, 18 grains are active and the number of partitions reduces to six;  
 242 in the final time step, only two partitions are needed. As grain elimination takes place, the number



**Fig. 5 Spatio-temporal generalization.** **(a)** Microstructure evolution for  $p_I$ :  $G = 2.4\text{K}/\mu\text{m}$ ,  $R = 1.52\text{m/s}$ ,  $\epsilon_k = 0.08$ . **(b)** The corresponding microstructure with doubled domain width with  $N_G = 8$ . **(c)** Microstructure for a longer time horizon  $t = 2t_H$ . **(d)** Microstructure for  $N_G = 8$  and doubled domain width and time horizon.



**Fig. 6 Generalization with number of grains.** (a) A 14-grain system is partitioned to four basic eight-grain fictitious systems (training configuration). Each basic GrainNN has a six-grain overlap with its left GrainNN and six-grain overlap with its right GrainNN. A single GrainNN iteration is taken independently for all subgroups; the middle two grains are kept from each subgroup followed by a repartitioning to account for grain elimination. (b) Time-dependent partitioning of a 32-grain system. In the presence of grain elimination, the number of partitions decreases as the number of active grains decreases. (c) An example of 32-grain prediction using GrainNN for testing parameter  $p_I$ . (d) An example of 128-grain prediction using GrainNN for testing parameter  $p_{II}$ . The entire system is decomposed to a set of 32-grain blocks (e.g., the orange and black blocks) with 16 grains shared with the adjacent blocks. Each 32-grain block is evolved independently using the single-level decomposition technique.



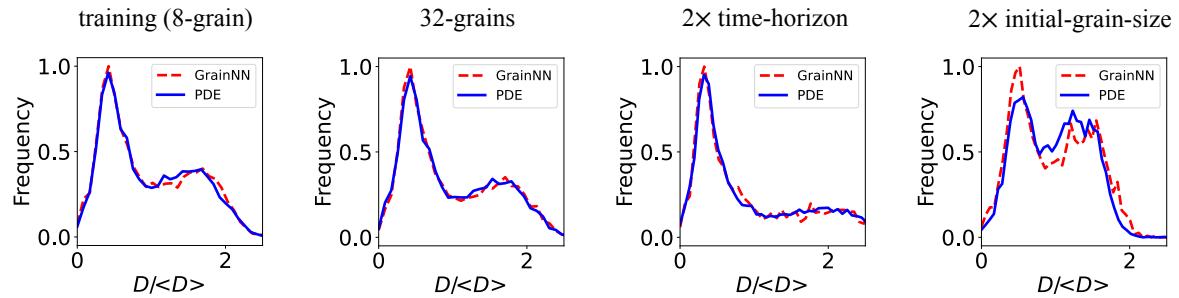
**Fig. 7 Non-rectangular melt pool generalization.** We show an example of a 32-grain spot weld melt pool with  $G$ ,  $R$  in the radial direction and a large number of grains. We use six steps to generate microstructures in such a geometry. Step 1: Initialize grains on a circular melt pool. The location of grain  $i$  is identified by  $\beta_i$ , the angle between the  $y$ -axis and the interface normal at the grain center. Step 2: Unroll grains to a flat interface using a coordinate transformation. The width of each substrate grain is preserved. Step 3: rotate  $G$  to the vertical direction, and rotate each grain by  $\beta_i$  to preserve the melt-pool misorientation angle between each grain and temperature gradient  $G$ . Step 4: evolve GrainNN with a height-dependent domain scaling factor. Step 5: rotate grains back to their original orientation. Step 6: invert the coordinate transformation to map back to melt pool geometry. Comparing with phase field simulation shows that GrainNN achieves about 9% MR accuracy.

of partitions decreases but the width of each partition  $L_x$  increases. We already show GrainNN is size-generalizable, so each partition can perform its own domain-size generalization with a time-dependent  $c_l$  after every GrainNN iteration. With this adaptive partition-assembly strategy, each partition is able to interact with grains in neighboring partitions throughout the inference. Fig. 6c is an example of 32-grain prediction with testing parameter  $p_I$ . The mean MR of the 32-grain setup is 8.3%, mostly caused by their time-dependent size generalization.

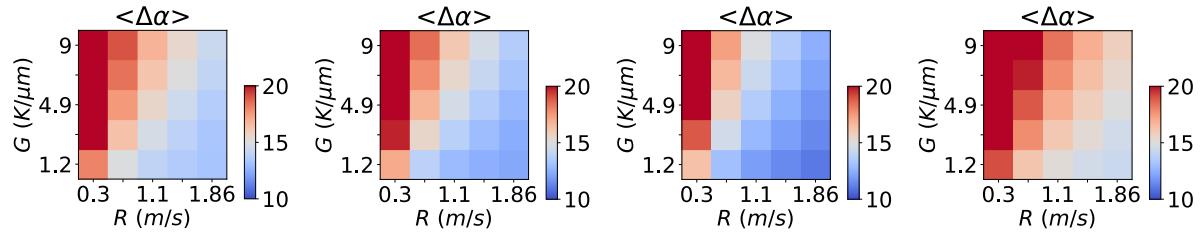
In principle, the process described above works for any number of grains. However, we've observed that the width-normalization step introduces significant errors for a larger number of grains. So we opt for a two-level decomposition: If the full system contains more than 32 grains (see Fig. 6d), we partition it into overlapping 32-grain blocks. Each block has a 16-grain overlap with its left neighbor and a 16-grain overlap with its right neighbor. Similar to the subgroups defined in the previous paragraph, each 32-grain block evolves independently but this time for several iterations. At the end of each independent run, we retain the 16 middle grains from each block. In summary, we use a two-level domain decomposition: the number of active grains at the SLI is first partitioned to 32-grain blocks with 16-grain left/right overlap; and each 32-grain block is partitioned to the eight-grain subgroups with 6-grain left/right overlap. We have not optimized the block/subgroup sizes; of course the greater the number of grains included in the base GrainNN the more accurate the approximation, but the FOM-related cost to generate training data will increase.

**Geometry:** Fig. 7 summarizes and demonstrates our scheme for non-rectangular geometries. We initialize 32 grains on a quarter circle and set  $G$  and  $R$  in the radial direction to mimic a melt pool setup. (We use a quarter circle by symmetry arguments.) The orientation of the temperature

(a) GrainNN-PDE QoI comparison for different configurations



(b) GrainNN QoI inference for the same configurations



**Fig. 8 Quantities of interest accuracy.** (a) Distribution of normalized grain size for testing parameter  $p_I$ :  $G = 2.4\text{K}/\mu\text{m}$ ,  $R = 1.52\text{m}/\text{s}$ ,  $\epsilon_k = 0.08$ . From left to right, the four configurations correspond to the in-distribution 8-grain setup, a 32-grain rectangular setup, and eight-grain setups with doubled time horizon and initial grain size. The blue lines are results of phase field simulations with 8,000 grains sampled for each configuration. The red dash-lines are corresponding GrainNN predictions with the same number of grains. (b) Expectations of area-weighted misorientation in a  $G$ - $R$  map for the four configurations. Kinetic anisotropy  $\epsilon_k = 0.13$ . All runs here are done using GrainNN. Each point in the map is averaged across 8000 grains.

gradient at grain  $i$  is indicated by the angle  $\beta_i$ . We use a coordinate transformation to map the circular interface to a rectangular interface. Then we use the GrainNN multigrain extension described in the previous paragraph for 32-grain rectangular geometries as in Fig. 6. We set  $G$  to the vertical direction and rotate each grain by  $\beta_i$  to keep the same misorientation angle with respect to  $G$ . Notice that, the melt pool geometry exposes contractive interface length along with radial growth, which leads to stronger grain competition and quicker grain elimination compared to the rectangular geometry. To mimic the same dynamics, we introduce a width scaling  $c_l$  which is calculated from the melt pool shape  $c_l(h) = 1 - h/R_m$ , where  $h$  is the interface height tracked in the rectangular domain and  $R_m$  is the radius of the melt pool. After the inference step with GrainNN we need to transform the solution back to the original domain, we rotate grains back to their original orientations and map grains back to the melt pool geometry. The final microstructure is compared with a phase field simulation in Fig. 7. We observe larger error is pronounced near the final SLI. Yet, the overall grain microstructure is captured well. Fig. 7 and Fig. 6c are initialized with exactly the same grain sequence, and the misclassification rate of the melt pool case is 1% larger than the rectangular domain case. We repeat this experiment for 100  $\xi$  values, the mean and standard deviation of MR are  $10.5 \pm 2.0\%$ .

**QoI convergence:** We conclude this section by discussing QoI accuracy for different out-of-distribution generalization scenarios. Fig. 8 summarizes our results. To establish a baseline, the first column shows in-distribution errors; the other three columns are 32-grain, doubled-time-horizon, and doubled-initial-grain-size configurations respectively. The top row shows distributional mismatches for  $p_I$ . The second row does not involve FOM runs, it just shows the mean QoI predictions using GrainNN for different  $p$  values. Regarding the first row results, the distribution generated by GrainNN is in excellent agreement with the phase field results. The corresponding KS statistics  $\kappa_D$  for the four subplots are 0.017, 0.012, 0.020, and 0.061, respectively. We can see that the distribution curve is less sensitive to the number of grains and more sensitive to the time horizon and initial

288 grain size. Regarding the second row results, every heatmap point is an average across GrainNN  
289 runs. We can observe that for the doubled-time-horizon configuration, the map is blue-shifted (more  
290 grains aligned with the temperature gradient); while for the doubled-initial-grain-size configuration,  
291 the map is redshifted.

### 292 3 DISCUSSION

293 We presented an accurate and efficient reduced-order model for microstructure evolution in 2D epi-  
294 taxial grain formation. By introducing grain features focused on capturing the solid-liquid interface,  
295 we achieved a significant representational compression. We utilized sequence-to-sequence and one-  
296 to-sequence LSTM frameworks to accomplish full-time prediction of feature evolution. GrainNN,  
297 once trained, is able to generate microstructures from random initial grain size and orientation dis-  
298 tributions without invoking FOMS to jump-start the predictions. GrainNN was trained over a wide  
299 range of  $G$  and  $R$  expected in AM solidification [8, 58].

300 Our results suggest that GrainNN generalizes quite well in predicting both pointwise microstruc-  
301 ture evolution and the statistics of quantities of interest for a wide range of scenarios. Our in-  
302 distribution generalization errors over parameters and grain initial conditions are in the 2%–10%  
303 range, whereas the out-of-distribution errors are in the 5%–15% range. The latter is dominated by  
304 errors due to different domain sizes. GrainNN is not sensitive to different  $p$  values, time horizon,  
305 and the number of grains.

306 **Computational efficiency.** GrainNN is accurate on pointwise predictions while achieving signif-  
307 icant speedups over FOMS, even after accounting for the cost of generating the training data. As  
308 we discussed it takes about 40 hours to run eight-grain phase field simulations to generate training  
309 data and 11 hours to train several GrainNNs of which we select the top four for ensemble inference.  
310 Notice our phase field solver is highly optimized with GPU [56]. Once trained, the GrainNN can run  
311 on a single CPU core while achieving high performance (Supplementary Table 2). An eight-grain  
312 inference takes about 50 ms and feature-to-microstructure reconstruction takes about 10 ms; both  
313 can be further accelerated on GPU. The average time for an eight-grain FOM solution is about one  
314 minute, meaning 1000× speedup for GrainNN. Cost of GrainNN scales linearly with time horizon  $t_H$   
315 and doesn't increase with the grain size. GrainNN achieves close to linear scaling for a large number  
316 of grains (100–1000) with approximately 20–30ms/grain. Simulating a large number of grains is  
317 expensive for our FOM since we need more phase fields and larger domains. For example, the time  
318 for a 32-grain phase field solve jumps to 36 minutes. If we were to compute QoI statistics (say using  
319 100  $\xi$  samples) for 100  $p$  for the 32-grain configuration, we would require 10,000 32-grain phase field  
320 runs, which would take up to thousands of hours (1 GPU). With GrainNN the calculations can be  
321 completed in 20 minutes on a multicore CPU. Thus, to compute multiparametric statistics, GrainNN  
322 is orders of magnitude faster than FOMS even with the data-generation cost.

323 **Applicability to different alloy systems.** We have examined the GrainNN accuracy for our  
324 model material stainless steel 316L. The same framework can be applied to other additive manufac-  
325 turing alloys such as Ti-6Al-4V, Al-Cu, and so on. The phase field model [57] we use to simulate  
326 grain microstructures neglects the solute diffusion and microsegregation, and purely considers the  
327 evolution of a polycrystal consisting of many crystal orientations. This model is valid for alloys whose  
328 melting range is small, for example, Ti-6-4, or whose absolute stability velocity [61] is lower than the  
329 pulling speed of the heat source. 316L stainless steel and Al-0.1 wt%Cu are in this regime for pulling  
330 speed in the range of 0.2–2 m/s. Phase field microstructures characteristics for 316L, Ti-6Al-4V,  
331 and Al-0.1wt%Cu are similar, based on their comparable phase field parameters (Supplementary  
332 Table 4). We repeated the same model training and testing procedures for Al-Cu, and the average  
333 misclassification rate for Al-Cu is 4.5% compared to 3.7% for stainless steel. For circumstances when  
334 solute diffusion and constitutional effects are prominent, phase field methods [56, 62, 63] resolving  
335 cell- or dendrite-scale microstructures should be used. In this case, the grain boundary movements  
336 can still be captured and reproduced by GrainNN. As we mentioned before GrainNN doesn't rely  
337 on the specifics of the FOM used to generate the training data. Any method capable of capturing  
338 the SLI dynamics and grain boundary evolution can be used to train GrainNN. It's also possible to

339 extract these features from experimental data [58, 64, 65].

340 **Applications and significance.** An important application of this work is the combination of a  
341 GrainNN-style ROM with crystal plasticity simulations to predict the localized mechanical properties  
342 within an AM component. The grain structure plays an important role in the mechanical proper-  
343 ties of AM materials [66]. In some cases it is the only important microstructural feature of note  
344 (e.g., when a narrow solidification range, a high solidification velocity, or congruent melting inter-  
345 metallic phase lead to minimal microsegregation). In other cases the grain structure is one of the  
346 multiple important features (variations in solid-solution strengthening, secondary phase formation,  
347 and dislocation densities) [66, 67]. The effect of the grain structure on the post-build mechanical  
348 properties of the grain structure can be calculated using crystal plasticity models [11, 68–73] (adding  
349 in the effects of other microstructural features using other models as needed [74]). These local-  
350 ized mechanical properties can be passed as constitutive relations for use in microstructure-aware  
351 component-scale simulations [68]. Therefore, the ability to efficiently create large ensembles of grain  
352 structures with GrainNN can be used to create representative grain structures to pass to crystal plas-  
353 ticity simulations. The ability of GrainNN to create specific realizations of grain structures (rather  
354 than simply statistical Qols) is particularly relevant for crystal plasticity models where an explicit  
355 representation of the grain structure is used to capture the detailed interactions between neighboring  
356 grains (e.g., plasticity models [11, 68, 69, 73] and viscoplasticity models [70, 72]). The application  
357 of such explicit-microstructure crystal plasticity simulations to predict constitutive relations for AM  
358 materials has already been demonstrated for grain structures generated by cellular automata [11, 68],  
359 phase field [71, 72], kinetic Monte Carlo [73], and experiments [70]; the ROM approach taken here  
360 with GrainNN can replace the microstructure generation portion of these workflows to accelerate the  
361 prediction of microstructure-aware mechanical properties.

362 The accurate, high-throughput grain structure predictions demonstrated here enable a higher-  
363 fidelity alternative to simplified analytic models for determining process conditions for an optimized  
364 grain structure within a part. One of the heralded potential benefits of AM is the ability to locally  
365 tune the microstructure for different, optimized properties across the component [75, 76]. Localized  
366 control of grain structures in 2D has been demonstrated by creating extreme changes in columnar  
367 versus equiaxed grain structures to form images such as the letters “DOE” [77], an oak leaf [78], and  
368 the Mona Lisa [79] using analytic-model-based heuristics for the favored grain structure using  $G$  and  
369  $R$  conditions from a thermal model. Similar control of grain orientation has been demonstrated to  
370 print QR codes into an AM material [80]. While extreme changes in microstructure can yield striking  
371 images, more subtle changes and 3D patterns are more likely to be of use in most applications.  
372 The current focus on demonstration builds and 1D [75, 81] or 2D structures [77–80] is a reflection  
373 of the difficulty in the transition to practical components. GrainNN has the potential to produce  
374 high-fidelity predictions of the grain structure that takes into account the effect of the underlying  
375 grains as the component is built layer by layer. ROMs could replace analytic-model-based heuristics  
376 in the scan strategy optimization while remaining computationally tractable.

377 **Limitations and extensions.** The two major limitations of the current version of GrainNN are:  
378 (1) ignoring nucleation and (2) two-dimensional simulations. For nucleation, we plan to introduce  
379 another network that maps a nucleus in the liquid to a new grain emerging on the SLI. GrainNN  
380 already handles systems of a time-varying number of grains. Once a new grain is formed at the SLI,  
381 we will reset GrainNN and capture future grain competition. Extending GrainNN to 3D will require  
382 richer features and more complex feature-to-microstructure maps. But we believe that basic features  
383 of GrainNN such as domain-decomposition and the sequence-to-sequence NA-LSTM architecture  
384 will be retained. In space, the NA-LSTM layer needs to find the neighbors of a grain in a 2D plane.  
385 Mathematically, it is easy to extend the current 1D-attention mechanism to 2D. The nontrivial part  
386 of learning 3D grain growth is what features can be used to describe the evolving microstructures  
387 and how to reconstruct microstructures from these features.

388 Besides nucleation and 3D there are several additional extensions. One is improving the interface  
389 representation by adding more features. Currently we’re using a rectangular representation by the  
390 width of the grain and the excess area and ignore higher-order interface shape details such as inter-  
391 face curvature. Sampling excess area is sufficient for computations of grain size and area-weighted  
392 misorientation. But if reconstruction of an accurate interface shape is required, more variables de-

scribing the above-interface microstructures can be fed into GrainNN to learn. A second extension is to coupling GrainNN to the data generation process drawing ideas from active learning [82, 83]. Currently, we sample  $G$ ,  $R$ , and  $\epsilon_k$  in a predetermined regular grid and selections of the parameter values are based on physical intuition. We expect a larger prediction error occurs where grain competition is stronger thus more parameters are sampled with high  $R$ ,  $\epsilon_k$  and low  $G$ . To minimize the number of phase field simulations to train the surrogate model, an adaptive method needs to be developed to find the optimal sampling locations. Another important extension is time-varying  $G(t)$  and  $R(t)$ , where an increasing  $R$  is frequently observed during the cooling process of a melt pool. If input signals of  $G(t)$  and  $R(t)$  are smooth, for instance a linearly increasing profile for  $R(t)$  [56, 58], the current ROM framework is directly applicable with a time sequence of  $G$  and  $R$  as the network input. However, if signals of  $G$  and  $R$  have steep gradients, using only coarse and equispaced feature sampling in the time dimension may omit dynamics happen between two snapshots. In this case we may need to change some architectural aspects of GrainNN.

Despite these limitations, GrainNN adopts several general ideas: The definition of a base ROM; the use of deep sequence-to-sequence LSTMs; the use of manually engineered SLI-based features; the use of a neighbor-aware transformer architecture to capture time-varying grain interactions; the hierarchical domain-decomposition and domain mapping ideas; and the physics-based network scaling to enhance generalization. We contend that these ideas extend to three-dimensional problems with nucleation and more complex material and process conditions. This is ongoing work in our group.

## 4 METHODS

### 4.1 Phase field solver.

We adopt the thermal multicomponent phase field model described in [57] but without nucleation. We assume a discrete orientation model in which each grain has one of  $N_G$  different crystalline orientations. Each phase field component  $\phi_\alpha$  is associated with an orientation  $\alpha_0$ . The dynamics of  $\phi_\alpha$  is governed by:

$$\begin{aligned} \tau_\alpha \frac{\partial \phi_\alpha}{\partial t} &= \nabla \cdot (W_\alpha^2 \nabla \phi_\alpha) + \sum_{j=x,y} \partial_j \left[ |\nabla \phi_\alpha|^2 W_\alpha \frac{\partial W_\alpha}{\partial (\partial_j \phi_\alpha)} \right] + \phi_\alpha - \phi_\alpha^3 - \lambda(1 - \phi_\alpha^2)^2 \frac{T - T_M}{L/C_p} \\ &\quad - \omega \frac{\phi_\alpha + 1}{2} \sum_{\beta \neq \alpha} \left( \frac{\phi_\beta + 1}{2} \right)^2, \quad \text{in } (Lx, Ly) \times (0, t_H), \quad \alpha = 1, \dots, N_G, \end{aligned} \quad (4.1)$$

where

$$\begin{aligned} \tau_\alpha &= \tau_0 (1 + 3\epsilon_k - 4\epsilon_k(\nabla_x \phi_\alpha^4 + \nabla_y \phi_\alpha^4)/|\nabla \phi_\alpha|^4), \\ T &= T_M + G(y - y_0 - Rt). \end{aligned} \quad (4.2)$$

Here  $L_x$ ,  $L_y$  are the domain dimensions;  $t_H$  is the time horizon;  $\tau_0$  is the interface attachment time scale;  $T_M$  is the melting temperature;  $y_0$  is the initial interface location;  $W_\alpha$  is the interface width;  $\lambda$  is the thermal coupling constant;  $L$  is the latent heat;  $C_p$  is the heat capacity;  $\omega$  is an interaction parameter that sets the repulsive strength between adjacent grains of different orientation [6]. We assume a four-fold kinetic anisotropy and due to the rapid solidification [84] neglect the capillary anisotropy. We also assume a given temperature field with constant  $G$  and  $R$  in the  $y$  direction. We use no-flux boundary conditions at the four domain boundaries. Material parameters, solver details, and convergence results are reported in the Supplementary (Note 4/Figure 5).

The initial condition for  $\phi_\alpha$ , i.e., the substrate-grain width, is sampled from a Gaussian distribution with mean  $w_\mu$  and variance  $w_\sigma^2$ . We set  $L_x = N_G w_\mu$ . The orientation of  $\phi_\alpha$  is sampled from the uniform distribution in  $[0, 90^\circ]$ . We set  $y_0 = 2\mu\text{m}$  and  $t_H = 60\mu\text{m}/R$  so that the final melting temperature isotherms for all simulations are at the same height  $62\mu\text{m}$ .

431 As discussed in the *Results* section, from each simulation we extract microstructure features  
 432  $\{\mathbf{z}_{i,t}\}_{i=1,t=0}^{N_G,T}$  at  $T$  time snapshots and parameters  $\{\boldsymbol{\theta}_i\}_{i=1}^{N_G}$ . These are defined as follows:

$$\begin{aligned}\mathbf{z}_{i,t} &= \begin{bmatrix} \Delta h_t & \Delta w_{i,t} & w_{i,t} & s_{i,t} \end{bmatrix}^T, \quad i = 1, \dots, N_G, \quad t = 0, \dots, T; \\ \boldsymbol{\theta}_i &= \begin{bmatrix} w_{i0} & \alpha_{i0} & \epsilon_k & G & R \end{bmatrix}^T, \quad i = 1, \dots, N_G.\end{aligned}\tag{4.3}$$

433 Here  $\Delta h_t = h_t - h_{t-1}$  is the height increment of the solid-liquid interface;  $\Delta w_{i,t} = w_{i,t} - w_{i,t-1}$  is  
 434 the width change for grain  $i$ ;  $s_{i,t}$  is the excess area above  $h_t$ .  $\boldsymbol{\theta}_i$  includes initial the grain width  $w_{i0}$   
 435 and orientation  $\alpha_{i0}$ , and repeated parameters  $\epsilon_k$ ,  $G$ , and  $R$ . (Notice that  $\boldsymbol{\theta}_i = (\xi_i, \mathbf{p})$ .) There are  
 436 obvious redundancies in this representation but empirically we found that having all of them improves  
 437 performance. In particular,  $\Delta h$  and  $\Delta w$  improve the scaling in the training loss function Eq. (4.7).  
 438 We did not use  $h$  and  $w$  in the loss function because these can vary significantly as  $t$  increases and  
 439 this can cause biased training for larger  $h$  and  $w$ .  $\Delta h$  and  $\Delta w$  have much smaller range. But we  
 440 still use  $w_{i,t}$  to track whether grain  $i$  is still active at time  $t$ . (Recall a grain is active when  $w_{i,t} > 0$ .)  
 441 We non-dimensionalize  $w_{i,t}$  with the  $L_x$  so that  $\sum_i w_{i,t} = 1$ . Finally, each element in  $\mathbf{z}_{i,t}$  and  $\boldsymbol{\theta}_i$  is  
 442 normalized to  $[-1, 1]$ .  $T$  is treated as tunable hyperparameter decided in GrainNN training.

443 Given  $\{\mathbf{z}_{i,t}\}_{i=1}^{N_G}$  and the microstructure at  $t - 1$ , we use the following steps to reconstruct the  
 444 microstructure at  $t$ : (i) Set  $h(t) = h(t - 1) + \Delta h(t)$ . (ii) Create grain-to-grain boundaries by accu-  
 445 mulating grain widths left to right. For instance, the grain boundary  $x$  position between the second  
 446 and third grain is at  $L_x w_{1,t} + L_x w_{2,t}$ . (iii) Interpolate grain boundary coordinates at every  $y$  between  
 447  $h(t - 1)$  and  $h(t)$ . (iv) Add a rectangle on the top of each grain with height  $\delta_i = s_{i,t}/w_{i,t}$ .

## 448 4.2 GrainNN architecture

449 The input to GrainNN at time  $t$  is a matrix  $X_t \in \mathbb{R}^{D_{in} \times N_G}$  defined by

$$X_t = \begin{bmatrix} \mathbf{x}_{1,t} & \mathbf{x}_{2,t} & \dots & \mathbf{x}_{N_G,t} \end{bmatrix}, \quad \mathbf{x}_{i,t} = \begin{bmatrix} \mathbf{z}_{i,t} \\ \boldsymbol{\theta}_i \\ t \end{bmatrix},\tag{4.4}$$

450 where  $D_{in} = 10$ . The output of GrainNN is  $Z_t = [\mathbf{z}_{1,t} \ \mathbf{z}_{2,t} \ \dots \ \mathbf{z}_{N_G,t}]$  for  $t = 1, \dots, T$ . Therefore  
 451 we write  $\{Z_t\}_{t=1}^T = \text{GrainNN}(X_0)$ , for eight-grain rectangular domains.

452 The basic structure of GrainNN is defined in Algorithm 1. In lines 2–5, the one-to-sequence  
 453 NA-LSTM  $\mathcal{N}_0$  is applied to  $X_0$  to generate the first  $J$  snapshot. In line 4,  $\mathcal{L}$  is an NA-LSTM layer  
 454 that computes the hidden and cell states  $H_t$  and  $C_t$  given their values at  $t - 1$  (and  $X_{t-1}$ ). The  
 455 actual GrainNN contains multiple  $\mathcal{L}$  layers (Supplementary Algorithm 1).  $\mathcal{L}$  is defined as follows.

$$\begin{aligned}i_t &= \sigma(A_{in}([X_{t-1}, H_{t-1}]) + W_{ci}C_{t-1} + b_i), \\ f_t &= \sigma(A_f([X_{t-1}, H_{t-1}]) + W_{cf}C_{t-1} + b_f), \\ C_t &= f_t C_{t-1} + i_t \tanh(A_c([X_{t-1}, H_{t-1}]) + b_c), \\ o_t &= \sigma(A_o([X_{t-1}, H_{t-1}]) + W_{co}C_t + b_o), \\ H_t &= o_t \tanh(C_t).\end{aligned}\tag{4.5}$$

Here  $W_{ci}, b_i, W_{cf}, b_f, W_{co}, b_o$  are the layer trainable parameters or “weights” in machine learning lingo. Here  $A_{in}$ ,  $A_f$ ,  $A_c$ ,  $A_o$  are nonlinear “attention” operators defined in the next section which also have trainable weights.  $H_0 \in \mathbb{R}^{D_h \times N_G}$  and  $C_0 \in \mathbb{R}^{D_h \times N_G}$  are initialized to zero;  $D_h$ , the layer size, is a hyperparameter. Given  $H_t$ , in line 5 we obtain  $Z_t$  by an output layer  $\mathcal{O}$  (and use it to define

$X_t$  using Eq. (4.4)). The components of the  $i$ th column of  $Z_t$ ,  $\mathbf{z}_{i,t} = \mathcal{O}_i(H_t)$ , are defined by:

$$w'_{i,t} = \text{ReLU}\left(w_{i,t-1} + (W_{hw}H_t + b_w)_i / c_l\right), \quad (4.6a)$$

$$w_{i,t} = \frac{w'_{i,t}}{\sum_i w'_{i,t}}, \quad (4.6b)$$

$$\Delta w_{i,t} = w_{i,t} - w_{i,t-1}, \quad (4.6c)$$

$$\Delta y_t = \text{ReLU}\left(W_{hy}H_t + b_y\right), \quad (4.6d)$$

$$s_{i,t} = \text{ReLU}\left(W_{hs}H_t + b_s\right)_i. \quad (4.6e)$$

456  $W_{hw}$ ,  $W_{hy}$ ,  $W_{hs}$  and  $b_w, b_y, b_s$  are trainable weights;  $c_l$  is the domain scaling factor ( $c_l = 1$  for training  
457 data). We use ReLU as an activation function since  $\Delta y_t, w_{i,t}, s_{i,t}$  are all non-negative. Eq. (4.6b) is  
458 a normalization step to ensure the sum of  $w_{i,t}$  is 1.

459 The second component of GrainNN, denoted by  $\mathcal{N}$ , is a sequence-to-sequence NA-LSTM comprising an “encoder”  $\mathcal{E}$  and a “decoder”  $\mathcal{D}$ . These two in turn, are composed of layers  $\mathcal{L}$  and  $\mathcal{O}$   
460 of the same architecture as  $\mathcal{N}_0$ . In lines 10–12 of Algorithm 1,  $\mathcal{E}$  compiles the latest  $J$  snapshots  
461 sequentially to  $H_{J-1}$  and  $C_{J-1}$ . In lines 15–17,  $\mathcal{D}$  takes  $H_{J-1}$  and  $C_{J-1}$  and predicts the next  $J$   
462 snapshots. This iteration is repeated for all  $t$ .  
463

---

**Algorithm 1** Architecture of GrainNN

Input:  $X_0$ . Output:  $Z_t$  for  $t = 0, 1, \dots, T$

Hyperparameters:  $J$  is the length of input/output sequence,  $T$  is the number of time snapshots.

---

```

1: /*  $\mathcal{N}_0 : X_0 \rightarrow \{X_1, X_2, \dots, X_{J-1}\}$  */
2:  $H_0, C_0 = 0$                                      Initialize hidden and cell states of  $\mathcal{N}_0$ 
3: for  $t = 1 \rightarrow J-1$  do
4:    $[H_t, C_t] = \mathcal{L}(X_{t-1}, [H_{t-1}, C_{t-1}])$ 
5:    $Z_t = \mathcal{O}(H_t)$                                 /*  $Z_t \rightarrow X_t$  using Eq. (4.4) */
6: /*  $\mathcal{N} : \{X_0, X_1, \dots, X_{J-1}\} \rightarrow \{X_J, X_{J+1}, \dots, X_{2J-1}\}$  */
7:  $t^* = 0$ 
8: while  $t^* < T$  do
9:   /* Encoder:  $\mathcal{E} : \{X_0, X_1, \dots, X_{J-1}\} \rightarrow H_{J-1}, C_{J-1}$  */
10:   $H_0, C_0 = 0$                                      Initialize hidden and cell states of  $\mathcal{N}$ 
11:  for  $t = 0 \rightarrow J-1$  do
12:     $[H_t, C_t] = \mathcal{L}(X_{t^*+t}, [H_{t-1}, C_{t-1}])$ 
13:    /* Decoder:  $\mathcal{D} : \{X_{J-1}, H_{J-1}, C_{J-1}\} \rightarrow \{X_J, X_{J+1}, \dots, X_{2J-1}\}$  */
14:    for  $t = J \rightarrow 2J-1$  do
15:       $[H_t, C_t] = \mathcal{L}(X_{t^*+t-1}, [H_{t-1}, C_{t-1}])$ 
16:       $Z_t = \mathcal{O}(H_t)$                                 /*  $Z_t \rightarrow X_t$  using Eq. (4.4) */
17:     $t^* += J$ 

```

---

464 The network weights are computed by solving an unconstrained optimization problem with a  
465 loss function, which we discuss here. Let  $\{Z_t, \dots, Z_{t+J-1}\}$  be a feature sequence extracted from the  
466  $n$ th phase field simulation; and let  $\{\tilde{Z}_t, \dots, \tilde{Z}_{t+J-1}\}$  be the corresponding GrainNN sequence. Then  
467 assuming we have data from  $M$  phase field simulations and  $S = T - 2J + 1$  time sequences per  
468 simulation, the loss function  $\mathcal{F}$  is defined as follows:

$$\begin{aligned} \mathcal{F}_s^{(n)} &= \frac{1}{J} \sum_t^{t+J-1} (\Delta h_t - \Delta \tilde{h}_t)^2 + \frac{1}{J} \sum_t^{t+J-1} \frac{1}{N_G} \sum_{i=1}^{N_G} (\Delta w_{i,t} - \Delta \tilde{w}_{i,t})^2 + (s_{i,t} - \tilde{s}_{i,t})^2, \\ \mathcal{F} &= \frac{1}{M} \sum_{n=1}^M \frac{1}{S} \sum_{s=1}^S \mathcal{F}_s^{(n)}. \end{aligned} \quad (4.7)$$

469 We use a simple data augmentation trick to train  $\mathcal{N}_0$ : We treat every snapshot, besides the tail  $J$   
 470 ones, as an initial condition; and its subsequent  $J - 1$  snapshots as the predicted sequence from it.  
 471 Thus  $S$  for  $\mathcal{N}_0$  becomes  $T - J + 1$  instead of just one. To train GrainNN we used 60 training epochs  
 472 using the Adam optimizer [85] applied for the first 50 training epochs and then switched to vanilla  
 473 stochastic gradient descent [86] for the last ten epochs. The learning rate is set to 50% decay every  
 474 10 epochs. Hyperparameters  $J, T, D_h$ , the number of NA-LSTM layers, and the initial learning rate  
 475 are tuned using a grid search. The total number of weights in the multilayer GrainNN is about 300K.

### 476 4.3 Attention operators

Let us now discuss the nonlinear attention operators  $A_{in}, A_f, A_c, A_o$  used in Eq. (4.5) and in every  
 477 NA-LSTM layer in  $\mathcal{N}_0$  and  $\mathcal{N}$ . We will use  $A_f$  as an example; the structure of the other three  
 478 operators is identical. Let  $F = A_f(Y)$ , with  $Y = [X \ H] \in \mathbb{R}^{(D_{in}+D_h) \times N_G}$  and  $F \in \mathbb{R}^{D_h \times N_G}$ .  $A_f$   
 479 is parameterized by three matrices  $\{W_f^{(h)} \in \mathbb{R}^{D_h \times (D_{in}+D_h)}\}_{h \in \{-1,0,1\}}$  and a vector  $b_f \in \mathbb{R}^{D_h}$ . The  
 480 three  $h$  values represent a grain's coupling to its left neighbor ( $h = -1$ ), itself ( $h = 0$ ), and its right  
 481 neighbor ( $h = 1$ ), see Fig. 2b. We refer to these three components as the “attention” components.  
 482 Let  $F_{f,i} = A_{f,i}(Y) \in \mathbb{R}^{D_h}$  be the attention for grain  $i$ :<sup>1</sup>

$$F_i = A_{f,i}(Y) = \sum_{h=-1}^1 A_{f,i}^{(h)}(Y) + b_f, \quad (4.8a)$$

$$A_{f,i}^{(h)}(Y) = \sum_{j=1}^{N_G} \text{softmax}_j \left( B_{i,j}^{(h)}(Y) \right) W_f^{(h)} Y_j, \quad h \in \{-1, 0, 1\}. \quad (4.8b)$$

$B^{(h)} \in \mathbb{B}^{N_G \times N_G}$  measures the coupling strength between grain  $i$  and grain  $j$  for attention component  $h$ . The softmax function is used to select the strongest coupling for grain  $i$ .  $A_{f,i}^{(h)}$  sums the contributions of all grains  $j$  weighted by  $W_f^{(h)}$  and softmax( $B^{(h)}$ ). In this paper, we propose a model  $B$  based on the current microstructure to find the nearest active neighbors for each grain:

$$B_{i,j}^{(h)}(Y) = (1 - (1 - e_i)\delta_{ij}) \left( P_{i,j}^{(h)} - c_0(1 - e_i e_j) \right), \quad e_i = \mathbb{1}_{w_i > 0}, \quad \delta_{ij} = \mathbb{1}_{i=j}, \quad (4.9a)$$

$$P_{i,j}^{(h)} = c_1 (-|i - j + h| + N_G \cdot \mathbb{1}_{-h(i-j+h) \geq 0}), \quad h \in \{-1, 0, 1\}, \quad (4.9b)$$

where  $c_1 = 10$ ,  $c_0 = 2||P||$ ;  $\delta$  is the Kronecker delta function. If  $e_i = 1$ , then grain  $i$  is active; and if  $e_i = 0$ , then grain  $i$  is eliminated. We designed Eq. (4.9a) to enable GrainNN to capture grain couplings in the presence of grain elimination: (i) if  $e_i = 0$ ,  $B_{i,i}^{(h)} > B_{i,i \neq j}^{(h)}$ , which means grain  $i$  is uncoupled to other grains; (ii) if  $e_i = 1$  and  $e_{j \neq i} = 0$ ,  $B_{i,j}^{(h)} \approx -c_0$ , which is small, then grain  $i$  and  $j$  are uncoupled; (iii) if  $e_i = e_j = 1$ , then  $B^{(h)} = P^{(h)}$ .

$P^{(h)}$  captures the relative position information of grains. Let us use  $h = 1$  as an example. The first term in  $P^{(1)}$  is a symmetric function maximized at  $j = i + 1$ , which lets softmax( $B^{(1)}$ ) find the right neighbor of grain  $i$  automatically. Similar position encoding functions can be found in vision transformers [87]. The issue of having only the first term in our problem setting is that if grain  $i + 1$  is eliminated, grain  $i$  and  $i + 2$  are both maximizers; but only grain  $i + 2$ , if it is active, should be the choice for the right neighbor. The second term in Eq. (4.9b), therefore, is introduced to specify that only grains to the right of grain  $i$  are the candidates of the right neighbor. The upper diagonal entries of  $P^{(1)}$  are increased by  $N_G$  so that  $P_{i,j>i}^{(1)} > P_{i,j \leq i}^{(1)}$ . Notice that for any grain on the right boundary of the computational domain ( $i = N_G$  or its right neighbors are all eliminated),  $P_{i,i}^{(1)} > P_{i,j < i}^{(1)}$ , thus grain  $i$  itself is selected as its right neighbor. This means the model  $P^{(h)}$  is valid for physical systems with no-flux boundary conditions, but not for Dirichlet or periodic boundary

---

<sup>1</sup>Not to be confused with  $h$  for the SLI height. Superscript  $(h)$  represents one component of the attention which is referred as one "head" in transformer notation [52].

493 conditions. The analysis is the same for  $h = -1$ . For  $h = 0$ ,  $\text{softmax}(B)$  approximately equals  
494 the identity matrix. With Eq. (4.9a) and Eq. (4.9b), every grain is able to find its nearest active  
495 neighbors for all time snapshots  $t$ .

## 496 Data Availability

497 The phase field solver that generates the training data is available at [https://github.com/YigongQin/PF\\_cpp](https://github.com/YigongQin/PF_cpp).  
498

## 499 Code Availability

500 The source code of GrainNN is available at [https://github.com/YigongQin/ROM\\_NN\\_PF](https://github.com/YigongQin/ROM_NN_PF).

## 501 Acknowledgments

502 This material is based upon work supported by the U.S. Department of Energy, Office of Science,  
503 Office of Advanced Scientific Computing Research, Applied Mathematics program under Award  
504 Number DE-SC0019393, by the U.S. Department of Energy, National Nuclear Security Administration  
505 Award Number DE-NA0003969; and by NSF award NSF OAC 2204226; Any opinions, findings, and  
506 conclusions or recommendations expressed herein are those of the authors and do not necessarily  
507 reflect the views of the DOE and NSF. Computing time on the Texas Advanced Computing Centers  
508 Stampede system was provided by an allocation from TACC and the NSF.

## 509 Author Contributions

510 Y.Q. and G.B. conceived the main ROM ideas. Y.Q. discovered the NA-LSTM transformer archi-  
511 tecture; wrote and verified the codes for the polycrystalline phase field and GrainNN; and conducted  
512 all numerical experiments. S.D and B.R. defined the Qols, the phase-field parameter regime, and  
513 contributed in the design and interpretation of experiments. G.B. supervised the work. All authors  
514 contributed to the discussion and writing of the paper.

## 515 Competing Interests

516 The authors declare no competing financial or non-financial interests.

## 517 References

- 518 [1] Smith, J. *et al.* Linking process, structure, property, and performance for metal-based ad-  
519 ditive manufacturing: computational approaches with experimental support. *Computational*  
520 *Mechanics* **57**, 583–610 (2016).
- 521 [2] Wang, Z., Palmer, T. A. & Beese, A. M. Effect of processing parameters on microstructure and  
522 tensile properties of austenitic stainless steel 304L made by directed energy deposition additive  
523 manufacturing. *Acta Materialia* **110**, 226–235 (2016).
- 524 [3] Li, M. *et al.* Microstructure and mechanical properties of 308L stainless steel fabricated by  
525 laminar plasma additive manufacturing. *Materials Science and Engineering: A* **770**, 138523  
526 (2020).
- 527 [4] Pinomaa, T., Lindroos, M., Walbrühl, M., Provatas, N. & Laukkanen, A. The significance of  
528 spatial length scales and solute segregation in strengthening rapid solidification microstructures  
529 of 316L stainless steel. *Acta Materialia* **184**, 1–16 (2020).

- 530 [5] Steinbach, I. & Pezzolla, F. A generalized field method for multiphase transformations using  
531 interface fields. *Physica D: Nonlinear Phenomena* **134**, 385–393 (1999).
- 532 [6] Ofori-Opoku, N. & Provatas, N. A quantitative multi-phase field model of polycrystalline alloy  
533 solidification. *Acta Materialia* **58**, 2155–2164 (2010).
- 534 [7] Yang, M., Wang, L. & Yan, W. Phase-field modeling of grain evolutions in additive manufac-  
535 turing from nucleation, growth, to coarsening. *npj Computational Materials* **7**, 1–12 (2021).
- 536 [8] Chadwick, A. F. & Voorhees, P. W. The development of grain structure during additive man-  
537 ufacturing. *Acta Materialia* **211**, 116862 (2021).
- 538 [9] Rodgers, T. M., Madison, J. D. & Tikare, V. Simulation of metal additive manufacturing  
539 microstructures using kinetic Monte Carlo. *Computational Materials Science* **135**, 78–89 (2017).
- 540 [10] Gandin, C.-A. & Rappaz, M. A 3D cellular automaton algorithm for the prediction of dendritic  
541 grain growth. *Acta Materialia* **45**, 2187–2195 (1997).
- 542 [11] Rolchigo, M., Carson, R. & Belak, J. Understanding Uncertainty in Microstructure Evolution  
543 and Constitutive Properties in Additive Process Modeling. *Metals* **12**, 324 (2022).
- 544 [12] Tan, J. H. K., Sing, S. L. & Yeong, W. Y. Microstructure modelling for metallic additive  
545 manufacturing: A review. *Virtual and Physical Prototyping* **15**, 87–105 (2020).
- 546 [13] Miyoshi, E. et al. Ultra-large-scale phase-field simulation study of ideal grain growth. *NPJ  
547 Computational Materials* **3**, 1–6 (2017).
- 548 [14] Miyoshi, E. et al. Large-scale phase-field study of anisotropic grain growth: Effects of  
549 misorientation-dependent grain boundary energy and mobility. *Computational Materials Sci-  
550 ence* **186**, 109992 (2021).
- 551 [15] Chang, K., Chen, L.-Q., Krill III, C. E. & Moelans, N. Effect of strong nonuniformity in grain  
552 boundary energy on 3-D grain growth behavior: A phase-field simulation study. *Computational  
553 Materials Science* **127**, 67–77 (2017).
- 554 [16] Burke, J. & Turnbull, D. Recrystallization and grain growth. *Progress in metal physics* **3**,  
555 220–292 (1952).
- 556 [17] Hillert, M. On the theory of normal and abnormal grain growth. *Acta metallurgica* **13**, 227–238  
557 (1965).
- 558 [18] Holm, E. A., Hassold, G. N. & Miodownik, M. A. On misorientation distribution evolution  
559 during anisotropic grain growth. *Acta Materialia* **49**, 2981–2991 (2001).
- 560 [19] DeWitt, S., Rudraraju, S., Montiel, D., Andrews, W. B. & Thornton, K. PRISMS-PF: A general  
561 framework for phase-field modeling with a matrix-free finite element method. *npj Computational  
562 Materials* **6**, 1–12 (2020).
- 563 [20] Vondrouš, A., Selzer, M., Hötzter, J. & Nestler, B. Parallel computing for phase-field models.  
564 *The International journal of high performance computing applications* **28**, 61–72 (2014).
- 565 [21] Shimokawabe, T. et al. Peta-scale phase-field simulation for dendritic solidification on the  
566 TSUBAME 2.0 supercomputer. In *Proceedings of 2011 International Conference for High  
567 Performance Computing, Networking, Storage and Analysis*, 1–11 (2011).
- 568 [22] Shi, X., Huang, H., Cao, G. & Ma, X. Accelerating large-scale phase-field simulations with  
569 GPU. *Aip Advances* **7**, 105216 (2017).
- 570 [23] Hijazi, S., Stabile, G., Mola, A. & Rozza, G. Data-driven POD-Galerkin reduced order model  
571 for turbulent flows. *Journal of Computational Physics* **416**, 109513 (2020).
- 572 [24] Ohayon, R. & Soize, C. *Advanced computational vibroacoustics: reduced-order models and  
573 uncertainty quantification* (Cambridge University Press, 2014).
- 574 [25] Frangos, M., Marzouk, Y., Willcox, K. & van Bloemen Waanders, B. Surrogate and reduced-  
575 order modeling: a comparison of approaches for large-scale statistical inverse problems. *Large-  
576 Scale Inverse Problems and Quantification of Uncertainty* 123–149 (2010).

- 577 [26] LeGresley, P. & Alonso, J. Airfoil design optimization using reduced order models based on  
578 proper orthogonal decomposition. In *Fluids 2000 conference and exhibit*, 2545 (2000).
- 579 [27] Noack, B. R., Morzynski, M. & Tadmor, G. *Reduced-order modelling for flow control*, vol. 528  
580 (Springer Science & Business Media, 2011).
- 581 [28] Benner, P., Gugercin, S. & Willcox, K. A survey of projection-based model reduction methods  
582 for parametric dynamical systems. *SIAM review* **57**, 483–531 (2015).
- 583 [29] Cohn, R. & Holm, E. Unsupervised machine learning via transfer learning and k-means clustering  
584 to classify materials image data. *Integrating Materials and Manufacturing Innovation* **10**, 231–  
585 244 (2021).
- 586 [30] Luo, Q., Holm, E. A. & Wang, C. A transfer learning approach for improved classification of  
587 carbon nanomaterials from TEM images. *Nanoscale Advances* **3**, 206–213 (2021).
- 588 [31] Chowdhury, A., Kautz, E., Yener, B. & Lewis, D. Image driven machine learning methods for  
589 microstructure recognition. *Computational Materials Science* **123**, 176–187 (2016).
- 590 [32] Li, X. *et al.* A transfer learning approach for microstructure reconstruction and structure-  
591 property predictions. *Scientific reports* **8**, 1–13 (2018).
- 592 [33] Bostanabad, R. Reconstruction of 3D microstructures from 2D images via transfer learning.  
593 *Computer-Aided Design* **128**, 102906 (2020).
- 594 [34] Rixner, M. & Koutsourelakis, P.-S. Self-supervised optimization of random material microstruc-  
595 tures in the small-data regime. *npj Computational Materials* **8**, 1–11 (2022).
- 596 [35] Yang, Z. *et al.* Deep learning approaches for mining structure–property linkages in high contrast  
597 composites from simulation datasets. *Computational Materials Science* **151**, 278–287 (2018).
- 598 [36] Ma, W. *et al.* Image-driven discriminative and generative machine learning algorithms for  
599 establishing microstructure–processing relationships. *Journal of Applied Physics* **128**, 134901  
600 (2020).
- 601 [37] Farizhandi, A. A. K., Betancourt, O. & Mamivand, M. Deep learning approach for chemistry  
602 and processing history prediction from materials microstructure. *Scientific reports* **12**, 1–14  
603 (2022).
- 604 [38] Cang, R. *et al.* Microstructure representation and reconstruction of heterogeneous materials  
605 via deep belief network for computational material design. *Journal of Mechanical Design* **139**,  
606 071404 (2017).
- 607 [39] Goodfellow, I. *et al.* Generative adversarial nets. *Advances in neural information processing  
systems* **27** (2014).
- 609 [40] Yang, Z. *et al.* Microstructural materials design via deep adversarial learning methodology.  
610 *Journal of Mechanical Design* **140** (2018).
- 611 [41] Lee, X. Y. *et al.* Fast inverse design of microstructures via generative invariance networks.  
612 *Nature Computational Science* **1**, 229–238 (2021).
- 613 [42] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780  
614 (1997).
- 615 [43] Montes de Oca Zapiaín, D., Stewart, J. A. & Dingreville, R. Accelerating phase-field-based  
616 microstructure evolution predictions via surrogate models trained by machine learning methods.  
617 *npj Computational Materials* **7**, 3 (2021).
- 618 [44] Yang, K. *et al.* Self-supervised learning and prediction of microstructure evolution with convo-  
619 lutional recurrent neural networks. *Patterns* **2**, 100243 (2021).
- 620 [45] Yan, W. *et al.* Predicting 2D Normal Grain Growth using a Physics-Regularized Interpretable  
621 Machine Learning Model. *arXiv preprint arXiv:2203.03735* (2022).
- 622 [46] Basak, A. & Das, S. Epitaxy and microstructure evolution in metal additive manufacturing.  
623 *Annual Review of Materials Research* **46**, 125–149 (2016).

- 624 [47] Cao, Y., Bai, P., Liu, F. & Hou, X. Grain growth in IN718 superalloy fabricated by laser additive  
625 manufacturing. *Materials Science and Technology* **36**, 765–769 (2020).
- 626 [48] Yan, F., Xiong, W. & Faierson, E. J. Grain structure control of additively manufactured metallic  
627 materials. *Materials* **10**, 1260 (2017).
- 628 [49] Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to sequence learning with neural networks. In  
629 *Advances in neural information processing systems*, 3104–3112 (2014).
- 630 [50] Xingjian, S. *et al.* Convolutional lstm network: A machine learning approach for precipitation  
631 nowcasting. In *Advances in neural information processing systems*, 802–810 (2015).
- 632 [51] Li, Z., Gavriluk, K., Gavves, E., Jain, M. & Snoek, C. G. Videolstm convolves, attends and  
633 flows for action recognition. *Computer Vision and Image Understanding* **166**, 41–50 (2018).
- 634 [52] Vaswani, A. *et al.* Attention is all you need. In *Advances in neural information processing  
635 systems*, 5998–6008 (2017).
- 636 [53] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional  
637 transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- 638 [54] Bello, I., Zoph, B., Vaswani, A., Shlens, J. & Le, Q. V. Attention augmented convolutional  
639 networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3286–  
640 3295 (2019).
- 641 [55] Ramachandran, P. *et al.* Stand-alone self-attention in vision models. *arXiv preprint  
642 arXiv:1906.05909* (2019).
- 643 [56] Qin, Y., Bao, Y., DeWitt, S., Radhakrishnan, B. & Biros, G. Dendrite-resolved, full-melt-pool  
644 phase-field simulations to reveal non-steady-state effects and to test an approximate model.  
645 *Computational Materials Science* **207**, 111262 (2022).
- 646 [57] Pinomaa, T. *et al.* Process-Structure-Properties-Performance modeling for selective laser melting.  
647 *Metals* **9**, 1138 (2019).
- 648 [58] Pinomaa, T. *et al.* Phase field modeling of rapid resolidification of Al-Cu thin films. *Journal of  
649 Crystal Growth* **532** (2020).
- 650 [59] Sun, D., Asta, M., Hoyt, J., Mendelev, M. & Srolovitz, D. Crystal-melt interfacial free energies  
651 in metals: fcc versus bcc. *Physical Review B* **69**, 020102 (2004).
- 652 [60] Dice, L. R. Measures of the amount of ecologic association between species. *Ecology* **26**,  
653 297–302 (1945).
- 654 [61] Trivedi, R., Sekhar, J. & Seetharaman, V. Solidification microstructures near the limit of  
655 absolute stability. *Metallurgical Transactions A* **20**, 769–777 (1989).
- 656 [62] Tourret, D. & Karma, A. Growth competition of columnar dendritic grains: A phase-field study.  
657 *Acta Materialia* **82**, 64–83 (2015).
- 658 [63] Takaki, T. *et al.* Competitive grain growth during directional solidification of a polycrystalline  
659 binary alloy: Three-dimensional large-scale phase-field study. *Materialia* **1**, 104–113 (2018).
- 660 [64] Liu, H., Zhu, S., Abbott, T. B., Zhen, Z. & Nie, J.-F. Quasi-in-situ EBSD Study of the  
661 Microstructure and Texture Evolution During Static Recrystallization in an Extruded Mg-Mn-  
662 Ce Alloy. *JOM* 1–17 (2022).
- 663 [65] Mantri, S. *et al.* The effect of boron on the grain size and texture in additively manufactured  
664  $\beta$ -Ti alloys. *Journal of Materials Science* **52**, 12455–12466 (2017).
- 665 [66] DebRoy, T. *et al.* Additive manufacturing of metallic components—process, structure and prop-  
666 erties. *Progress in Materials Science* **92**, 112–224 (2018).
- 667 [67] Kok, Y. *et al.* Anisotropy and heterogeneity of microstructure and mechanical properties in  
668 metal additive manufacturing: A critical review. *Materials & Design* **139**, 565–586 (2018).
- 669 [68] Turner, J. A. *et al.* ExaAM: Metal additive manufacturing simulation at the fidelity of the  
670 microstructure. *The International Journal of High Performance Computing Applications* **36**,  
671 13–39 (2022).

- 672 [69] Roters, F. *et al.* Overview of constitutive laws, kinematics, homogenization and multiscale  
673 methods in crystal plasticity finite-element modeling: Theory, experiments, applications. *Acta  
674 materialia* **58**, 1152–1211 (2010).
- 675 [70] Sangid, M. D., Nicolas, A., Kapoor, K., Fodran, E. & Madsen, J. Modeling the Role of  
676 Epitaxial Grain Structure of the Prior  $\beta$  Phase and Associated Fiber Texture on the Strength  
677 Characteristics of Ti-6Al-4V Produced via Additive Manufacturing. *Materials* **13**, 2308 (2020).
- 678 [71] Lim, H. *et al.* Incorporating physically-based microstructures in materials modeling: Bridging  
679 phase field and crystal plasticity frameworks. *Modelling and Simulation in Materials Science  
680 and Engineering* **24**, 045016 (2016).
- 681 [72] Liu, P. *et al.* Integration of phase-field model and crystal plasticity for the prediction of process-  
682 structure-property relation of additively manufactured metallic materials. *International Journal  
683 of Plasticity* **128**, 102670 (2020).
- 684 [73] Rodgers, T. M., Lim, H. & Brown, J. A. Three-dimensional additively manufactured microstruc-  
685 tures and their mechanical properties. *JOM* **72**, 75–82 (2020).
- 686 [74] Aagesen, L. *et al.* Prisms: An integrated, open-source framework for accelerating predictive  
687 structural materials science. *JOM* **70**, 2298–2314 (2018).
- 688 [75] Popovich, V. *et al.* Functionally graded Inconel 718 processed by additive manufacturing:  
689 Crystallographic texture, anisotropy of microstructure and mechanical properties. *Materials &  
690 Design* **114**, 441–449 (2017).
- 691 [76] Tammas-Williams, S. & Todd, I. Design for additive manufacturing with site-specific properties  
692 in metals and alloys. *Scripta Materialia* **135**, 105–110 (2017).
- 693 [77] Dehoff, R. R. *et al.* Site specific control of crystallographic grain orientation through electron  
694 beam additive manufacturing. *Materials Science and Technology* **31**, 931–938 (2015).
- 695 [78] Halsey, W., Ferguson, J., Plotkowski, A., Dehoff, R. & Paquit, V. Geometry-independent mi-  
696 crostructure optimization for electron beam powder bed fusion additive manufacturing. *Additive  
697 Manufacturing* **35**, 101354 (2020).
- 698 [79] Plotkowski, A. *et al.* A stochastic scan strategy for grain structure control in complex geometries  
699 using electron beam powder bed fusion. *Additive Manufacturing* **46**, 102092 (2021).
- 700 [80] Sofinowski, K., Wittwer, M. & Seita, M. Encoding data into metal alloys using laser powder  
701 bed fusion. *Additive Manufacturing* **52**, 102683 (2022).
- 702 [81] Niendorf, T. *et al.* Functionally graded alloys obtained by additive manufacturing. *Advanced  
703 engineering materials* **16**, 857–861 (2014).
- 704 [82] Lookman, T., Balachandran, P. V., Xue, D. & Yuan, R. Active learning in materials science  
705 with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational  
706 Materials* **5**, 1–17 (2019).
- 707 [83] Krogh, A. & Vedelsby, J. Neural network ensembles, cross validation, and active learning.  
708 *Advances in neural information processing systems* **7** (1994).
- 709 [84] Bragard, J., Karma, A., Lee, Y. H. & Plapp, M. Linking phase-field and atomistic simulations  
710 to model dendritic solidification in highly undercooled melts. *Interface Science* **10**, 121–136  
711 (2002).
- 712 [85] Zhang, Z. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th  
713 International Symposium on Quality of Service (IWQoS)*, 1–2 (ieee, 2018).
- 714 [86] Goodfellow, I., Bengio, Y. & Courville, A. *Deep learning* (MIT press, 2016).
- 715 [87] Cordonnier, J.-B., Loukas, A. & Jaggi, M. On the relationship between self-attention and  
716 convolutional layers. *arXiv preprint arXiv:1911.03584* (2019).