AnisoGNN: Supplementary materials

1 Supplementary materials

1.1 Microstructure generation with Dream.3D

To build a training dataset for the GNN models considered in this study, we first generated digital 3D microstructure volume elements (MVEs). Our procedure of the MVE generation included two separate steps of (i) creating 3D grains (grain structure) and (ii) their crystallographic orientations (texture).

Grain structure. We used open-source software Dream.3D (Groeber et al., 2014) to generate 300 single-phase polycrystalline equiaxed MVEs. Each $128 \times 128 \times 128$ MVE (with voxel size of $1\,\mu\mathrm{m}^3$) contained more than 1000 grains. For each grain in an MVE, we assigned a crystallographic orientation sampled from separately generated crystallographic textures.

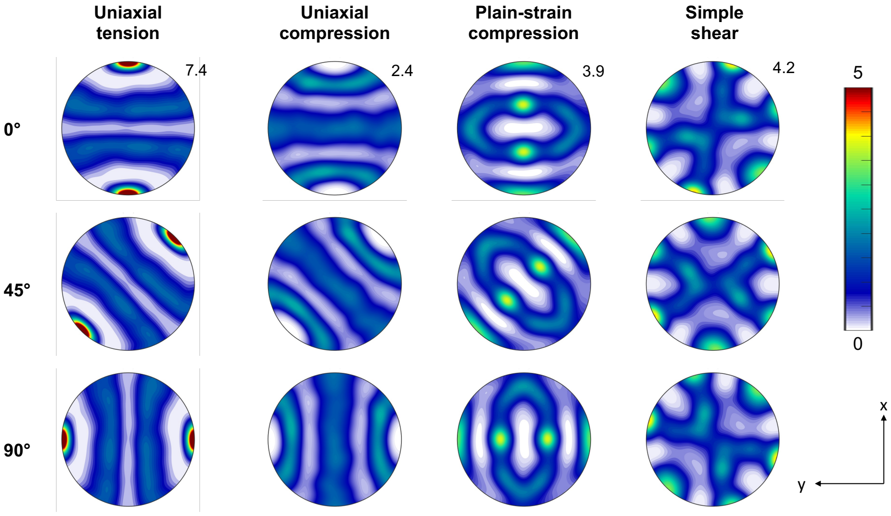

Figure 1: Pole figures (for (111) plane normals) showing 12 textures used for microstructure generation in the present study. The angle in degrees indicates the rigid rotation around the z axis (out of the figure plane). The number next to pole figures indicates the maximum intensity.

Crystallographic texture. We created multiple initial textures for MVEs to capture a diverse set of anisotropic mechanical properties. Textures were obtained by polycrystal plasticity simulations under different boundary conditions. We obtained four textures corresponding to uniaxial tension, uniaxial compression, plain strain compression, and simple shear deformation. We ran polycrystal plasticity simulations with these boundary conditions on a polycrystal consisting of 500 initially random crystal orientations. Our polycrystal plasticity simulations used Taylor homogenization scheme implemented in the open-source MTEX code (Bachman et al., 2010).

Following polycrystal plasticity simulations, we calculated orientation distribution functions (ODFs) from 500 discrete orientations of the deformed polycrystals. To further enrich the dataset, we obtained additional textures by rigid rotation of the four ODFs to 45° and 90° about the z axis. We therefore obtained 12 distinct ODFs, including the textures obtained by polycrystal plasticity simulations and their two rotated versions (Fig. 1). These 12 ODFs were used to sample discrete orientations for grains in the MVEs generated in Dream.3D as described above. For each of 12 textures, we allocated 25 MVEs with individual sets of discrete orientations sampled from the ODFs. The final microstructrue dataset contained 300 unique MVEs representing 12 initial textures (25 MVEs per texture) with more than 1000 grains in every MVE. Fig. 2 shows a few typical MVEs generated in this study. This microstructure dataset was used for micromechanical simulations of effective properties.

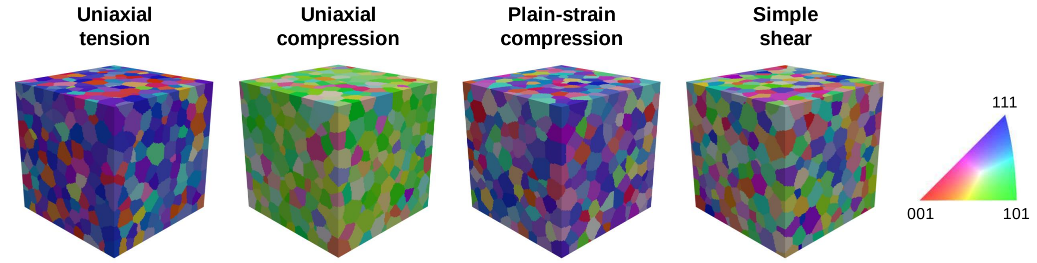

Figure 2: Typical MVEs for each of the four unrotated textures generated in this study. Grains in the MVEs are color-coded according to their crystallographic orientations and using the inverse pole figure colors.

1.2 Property simulations

To obtain effective properties of MVEs, we used fast Fourier transform (FFT) solver implemented within the Düsseldorf Advanced Material Simulation Kit (DAMASK) (Roters et al., 2019). We performed FFT simulations of uniaxial tension by applying mixed boundary conditions described by the rate of the deformation gradient and the first Piola–Kichhoff stress.

For the effective elastic modulus of René 88DT, we ran simulations to approx. $0.01\,\%$ total applied strain using purely elastic constitutive behavior of the material grid points. For the effective elastic modulus and yield strength of aluminum, we used elasto-plastic constitutive equations and MVEs were deformed to the total strain of $0.04\,\%$. For the plastic behavior, we adopted the power-law constitutive law with the following key model parameters: $\dot{\gamma}_0 = 0.001\,\mathrm{s}^{-1}$, $g_0 = 31\,\mathrm{MPa}$, $g_\infty = 63\,\mathrm{MPa}$, and $h_0 = 75\,\mathrm{MPa}$ (Eisenlohr et al., 2013).

1.3 Polycrystal graphs and graph neural networks

Polycrystal graphs. To allow GNNs learn microstructure–property relationships, we described 3D polycrystalline microstructures using undirected graphs. In these graphs, each node represents an individual grain, while the edges connect neighboring grains (with a shared boundary) taking periodicity in MVEs into account. Each grain-node in the graph was assigned a set of attributes, which included the grain volume (in voxels) and elements of either the quaternion vector describing the grain orientation (for GNN-O models), or rotated tensor relevant to the effective property of interest (for AnisoGNN models). The AnsioGNN-C models for effective elastic modulus relied on the 21 elements of the full elastic stiffness tensor. The AnisoGNN-S models for modeling effective yield strength used 108 (concatenated) elements of the 12 Schmid tensors corresponding to the 12 f.c.c. slip systems. Before being assigned to nodes as attributes, both the stiffness tensor and the Schmid tensors of each grain were rotated to the global frame using the grain orientations (Kalidindi et al., 1992).

GNN-O models. For GNN-O models discussed in the main text, we adopted the GNN architecture reported by Hestroffer et al. (Hestroffer et al., 2023). The attributes (quaternions) of the nodes sequentially pass through a pre-processing fully-connected layer, two message-passing layers with the SAGE convolution (Hamilton et al., 2017), a global mean pooling layer, two post-processing fully-connected layers (all with ReLU activation except global mean pooling) followed by an output layer. All GNN-O models were trained for 600 epochs.

AnisoGNN models. For AnisoGNN models, we designed a simple GNN architecture with graph convolutional layer followed by global mean pooling and a post-processing fully-connected layer for output. The graph convolutional layer, based on a mean message passing scheme, first applies a linear transformation (with ReLU activation) to the node attributes, then aggregates attributes from neighboring nodes by averaging (mean aggregation). The fully connected layer of the convolutional layer linearly transforms the input from 22 (grain size and 21 stiffness tensor elements) to six features in the AnisoGNN-C models and from 109 (grain size and 108 elements of 12 Schmid tensors) to 19 in the AnisoGNN-S models. Global mean pooling consolidates transformed node attributes of each graph into a single feature vector. The network concludes with a layer that maps this pooled graph representation to the output. All AnisoGNN models were trained for 800 epochs.

Code and data availability

The codes and data generated during the current study are available on GitHub at https://github.com/hugy888/AnisoGNN.