TECHNICAL ARTICLE

PRISMS-Plasticity TM: An Open-Source Rapid Texture Evolution Analysis Pipeline

Mohammadreza Yaghoobi[1] · John E. Allison[1] · Veera Sundararaghavan[2]

Abstract

The evolution of texture during different mechanical processes is an integral part of Integrated Computational Materials Engineering (ICME) of metals and alloys. In the present work, PRISMS-Plasticity TM, an open-source rapid texture evolution analysis pipeline, based on the Taylor model is integrated into the open-source crystal plasticity software, PRISMS-Plasticity. However, unlike PRISMS-Plasticity, the pipeline does not involve any finite element solver. The Taylor-type crystal plasticity model is used to capture the evolution of texture. The pipeline aims to significantly minimize the cost of texture evolution simulations, so one can use the framework along with data-driven ICME design methods and generate high-throughput data with low costs for general analysis or machine learning applications. All the rate-independent and rate-dependent crystal plasticity models available in the Materials library of PRISMS-Plasticity can be used in the pipeline. The models can capture extension twinning as an important deformation mechanism in alloys with hexagonal close-packed crystal structures. The PRISMS-Plasticity TM pipeline is efficiently integrated with experimental characterization techniques such as electron backscatter diffraction (EBSD), and these pipelines use other open-source software packages such as DREAM.3D where appropriate. In addition, the PRISMS-Plasticity TM results are integrated with the PRISMS Center Materials Commons information repository. In this investigation, three different examples of texture evolution in a polycrystalline OFHC copper, extruded Mg alloy ZK60A, and an aluminum alloy 7075-T6 are presented to show the capability of PRISMS-Plasticity TM. The pipeline can take advantage of the parallel performance of PRISMS-Plasticity and scales exceptionally well for large problems running on multiple processors. The computational efficiency of this tool can be used to build large databases for use in machine learning applications. Such a large dataset is established containing process–microstructure relationships of 7075-T6 Al. The extracted data is visualized in 2-D latent space using an unsupervised machine learning model called the variational autoencoder (VAE). The trained VAE model is used to identify economical process sequences which lead to desired properties.

Keywords Texture · Crystal plasticity · PRISMS-Plasticity · ICME · Twinning · Machine learning

Introduction

Various processes have been used to design alloys with superior properties. Polycrystalline metals and alloys consist of thousands of grains each having a specific orientation. The distribution of these crystallographic orientations, which is commonly known as the texture, is an important property of a material sample that can define its response to different loading conditions. Accordingly, the material can be designed to exhibit the target properties by modifying the texture using specific and sometimes complex mechanical processes (e.g., cross-rolling, or forging of extruded billets). Rapid analysis of the texture evolution for different loading processes can be an integral part of any Integrated Computational Materials Engineering (ICME) frameworks in the design of new structural alloys and forming processes. Various combinations of processes can be investigated to find out the process combination which leads to the target performance. Accordingly, the efficiency and cost of each texture evolution simulation is an important consideration especially during the initial search over the entire process space.

✉ Mohammadreza Yaghoobi
  yaghoobi@umich.edu

1 Materials Science and Engineering, University of Michigan, Ann Arbor, MI 48109, USA

2 Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Crystal plasticity is a common tool to capture texture evolution during different loadings. Different crystal plasticity schemes have been developed to capture the response of metals and alloys including Taylor model [1], modified Taylor model including relaxation [2–4], cluster models such as LAMEL [5, 6], GIA [7, 8], and RGC [9], viscoplastic self-consistent (VPSC) models [10], elastic–rigid–plastic self-consistent (EPSC) models [11], elastic–viscoplastic self-consistent (EVPSC) models [12], spectral finite elements schemes using fast Fourier transform [13, 14], and crystal plasticity finite elements method (CPFEM) [15]. Depending on the target of the crystal plasticity simulation, each of these methods could be efficient. The simplest model and fastest is the Taylor model [1]. Using the Taylor model, it is possible to neglect the elastic part of the response to increase the speed of the framework. However, this assumption will introduce errors in the very common situation in which the material design process includes unloading or changes in the loading path. Accordingly, one should include elastic deformation in the Taylor model to cover all potential loading patterns. VPSC [10] is another efficient model that can be used to model polycrystalline samples. However, the VPSC artificially predicts a discontinuous response in the cases of unloading or change in the loading paths since it does not include the elastic deformation. An alternative is to use EVPSC models [12] which can provide more accurate response compared Taylor model. EVPSC is computationally more expensive compared to the Taylor model. CPFEM is the most computationally costly method, but it is able to address complex aims such as the effect of the grain neighborhood, complex boundary conditions, and voids and porosity. In the initial phase of process design to reach a target performance, the cost of the simulation is very critical since a variety of process combinations should be investigated. Using a complex method like CPFE is not practical at this stage. The best practice is to narrow down the process design space using the fastest crystal plasticity scheme which is the Taylor model. At a later stage, one can use the CPFE to study more advanced properties such as defects, neighborhood, and dislocation density evolution. In addition, for the very high strains found in many manufacturing processes, the Taylor-type model does not face the convergence issues which are a common problem within finite element (FE) frameworks.

Machine learning (ML) techniques have been employed in different aspects of materials science and engineering [16]. ML frameworks have been developed to capture the process–microstructure–property in different applications such as additive manufacturing [17–19], processing path design [20], continuum materials mechanics [21], free energy calculations [22], and fracture mechanics [23]. Independent of the ML applications, it is necessary to acquire a large enough dataset to train and validate the framework.

Accordingly, high throughput data generation is an integral part of any ML framework. One of the recent applications of ML is manufacturing process path design. Components may undergo many complex processes to reach the target performance. Designing such complex process paths requires significant resources. ML can narrow down the options for multiple process designs by providing input to the designers. A rapid and parallel texture evolution analysis pipeline can be effectively integrated with ML frameworks to investigate complex process paths.

In the current work, PRISMS-Plasticity TM, an open-source pipeline, is integrated into the open-source CPFE software PRISMS-Plasticity [15] to rapidly capture and analyze the texture evolution of metallic samples subjected to mechanical processes. This pipeline can efficiently generate high-throughput data for ML applications applied to the process–microstructure–property problem. PRISMS-Plasticity is an open-source CPFE software that has been used for many applications including the multiscale twinning–detwinning model Mg alloys [24], capturing the deformation mechanisms extracted from the high energy diffraction microscopy (HEDM) experiment [25], effects of heat treatment on the response of WE43 Mg alloy [26], investigation of the twin and strain maps in WE43 and Mg-4Al Mg alloys extracted from SEM-DIC experiment [27, 28], and microstructural fatigue analysis [29–32]. Although PRISMS-Plasticity is a very efficient software and it scales well for multi-thread simulations, it is not feasible for a global search of different process combinations to optimize the specific response of metals and alloys. Incorporation of PRISMS-Plasticity TM into the PRISMS-Plasticity framework enables consistent and efficient use of materials model in both simple Taylor model simulations as well as more complex CPFE simulations. In addition, all the features available in PRISMS-Plasticity are available for use with PRISMS-Plasticity TM. In the pipeline described here, no FE simulation is required to capture the texture evolution. The current framework incorporates Taylor-type crystal plasticity models to minimize the cost of texture evolution analysis and is also able to reach very high strains. The updated orientations are passed to the MATLAB toolbox MTEX [33] to plot the pole figures, and the postprocessing MATLAB scripts are provided as a part of the pipeline. Both rate-independent and rate-dependent crystal plasticity models available in the Materials library of PRISMS-Plasticity can be used in the TM pipeline. In the case of many metals and alloys with hexagonal close-packed crystal structures, extension twinning occurs as an important deformation mechanism in addition to the general plastic slip. Twinning is commonly captured in crystal plasticity models as polar pseudo-slip systems [34]. Several twinning models have been developed and used in different crystal plasticity frameworks, which was reviewed by Yaghoobi et al. [34].

Extension twinning is modeled in PRISMS-Plasticity TM using the predominant twin reorientation scheme (PTR) scheme in which the grain is entirely reorientated according to the most active twin pseudo-system when it reaches a certain threshold [35]. The PRISMS-Plasticity TM pipeline is integrated with experimental characterization techniques such as electron backscatter diffraction (EBSD) using DREAM.3D [36] and MTEX toolbox [33].

As an important step for data availability, reusability, and reproducibility, the pipeline results are integrated with the PRISMS Center Materials Commons information repository. The texture evolution is simulated in polycrystalline OFHC copper, extruded ZK60A Mg alloy, and 7075-T6 Al alloy. Both strong-scaling and weak-scaling are investigated for the PRISMS-Plasticity TM, which shows that the pipeline scales very well with the number of processors and problem size. Finally, an application of the improved computational efficiency of this method coupled with a ML framework is demonstrated by investigating the process–microstructure space of polycrystalline Al alloy subjected to combinatorial complexity of a three-stage deformation process. A variational autoencoder is incorporated to visualize the data in 2-D latent space. As a last step, sequence identification is conducted for the trained ML model to predict sequences that result in similar properties.

Crystal Plasticity Model

A new rapid texture evolution pipeline is presented as a part of PRISMS-Plasticity software [15] which incorporates the Taylor-type crystal plasticity model. The core assumption of the model is that all grains within a sample will sustain the same deformation gradient tensor $\mathbf{F}$ independent of their orientation. The plastic and elastic deformation gradient tensors within each grain, however, evolve differently depending on their orientation, following the multiplicative decomposition as below (see, e.g., [37, 38]):

$$\mathbf{F} = \mathbf{F}^{e1}\mathbf{F}^{p1} = \mathbf{F}^{e2}\mathbf{F}^{p2} = \cdots = \mathbf{F}^{eI}\mathbf{F}^{pI} \tag{1}$$

where $\mathbf{F}^e$ and $\mathbf{F}^p$ are elastic and plastic deformation gradient tensors, respectively. The velocity gradient tensor can be defined as follows:

$$\begin{aligned} l &= \dot{\mathbf{F}}\mathbf{F}^{-1} = \left(\dot{\mathbf{F}}^e\mathbf{F}^p + \mathbf{F}^e\dot{\mathbf{F}}^p\right)\left(\mathbf{F}^{p-1}\mathbf{F}^{e-1}\right) \\ &= \dot{\mathbf{F}}^e\mathbf{F}^{e-1} + \mathbf{F}^e\dot{\mathbf{F}}^p\mathbf{F}^{p-1}\mathbf{F}^{e-1} \end{aligned} \tag{2}$$

Accordingly, the total velocity gradient tensor can be additively decomposed to the elastic and plastic components as follows:

$$l = l^e + l^p \tag{3}$$

where $l^e = \dot{\mathbf{F}}^e\mathbf{F}^{e-1}$ and $l^p = \mathbf{F}^e\dot{\mathbf{F}}^p\mathbf{F}^{p-1}\mathbf{F}^{e-1} = \mathbf{F}^e\mathbf{L}^p\mathbf{F}^{e-1}$ in which $\mathbf{L}^p$ is the isoclinic intermediate configuration can be obtained by compiling the shearing rates of different slip systems as follows:

$$\mathbf{L}^p = \dot{\mathbf{F}}^p\mathbf{F}^{p-1} = \sum_{\alpha=1}^{N_s+N_t} \dot{\gamma}^\alpha \mathbf{S}^\alpha \text{sign}(\tau^\alpha) \quad \text{Rate} - \text{independent} \tag{4a}$$

$$\mathbf{L}^p = \dot{\mathbf{F}}^p\mathbf{F}^{p-1} = \sum_{\alpha=1}^{N_s+N_t} \dot{\gamma}^\alpha \mathbf{S}^\alpha \quad \text{Rate} - \text{dependent} \tag{4b}$$

where $\dot{\gamma}^\alpha$ and τ^α are the shearing rate and resolved shear stress, respectively, and $\mathbf{S}^\alpha$ is the Schmid tensor in the intermediate configuration corresponding to the αth slip/twin system. The resolved shear stress on α^{th} slip/twin system can be obtained using the Cauchy stress tensor $\boldsymbol{\sigma}$ as below:

$$\tau^\alpha = \boldsymbol{\sigma} : \mathbf{S}^\alpha. \tag{5}$$

The governing equation of crystal plasticity models can be defined as rate-independent or rate-dependent. In the case of former one, a yield surface is defined for α^{th} slip/twin system using the resolved shear stress τ^α and slip resistance s^α as follows:

$$g^\alpha = |\tau^\alpha| - s^\alpha. \tag{6}$$

In the case of the rate-dependent model, the flow rule can be defined as follows:

$$\dot{\gamma}^\alpha = \dot{\gamma}_0 \left|\frac{\tau^\alpha - \chi^\alpha}{s^\alpha}\right|^m \text{sgn}(\tau^\alpha - \chi^\alpha) \tag{7}$$

where χ^α is the back stress for α^{th} slip/twin system, and $\dot{\gamma}_0$ and m are the reference shearing rate and inverse strain rate sensitivity exponent, respectively.

Isotropic hardening is included for both rate-independent and rate-dependent crystal plasticity formulations as follows:

$$\dot{s}^\alpha = \sum_\beta h^{\alpha\beta}\left|\dot{\gamma}^\beta\right| \tag{8}$$

where $h^{\alpha\beta}$ is the hardening moduli and can be defined as follows:

$$h^{\alpha\vartheta} = \begin{cases} h_0^\vartheta\left[1 - \frac{s^\vartheta}{s_s^\vartheta}\right]^{a^\vartheta} & \text{if } \alpha = \vartheta \text{ (coplanar systems)} \\ h_0^\vartheta q\left[1 - \frac{s^\vartheta}{s_s^\vartheta}\right]^{a^\vartheta} & \text{if } \alpha \neq \vartheta \end{cases} \tag{9}$$

where q, h_0^ϑ, and s_s^ϑ are the latent hardening ratio, initial hardening parameter, and saturation slip resistance, respectively. a^ϑ is a material constant governing the evolution of hardening moduli. The kinematic hardening is included only in the

rate-dependent crystal plasticity model and is evolved using a two-term back stress Ohno–Wang type as follows:

$$\dot{\chi}_i^\alpha = h_i \dot{\gamma}^\alpha - r_i \left(\frac{\left| \chi_i^\alpha \right|}{b_i} \right)^{m_i} \chi_i^\alpha |\dot{\gamma}^\alpha|, \quad \dot{\chi}^\alpha = \sum_{i=1}^{2} \dot{\chi}_i^\alpha \qquad (10)$$

where r_i and h_i are the material constants, $b_i = h_i/r_i$, and m_i governs the dynamic recovery.

In order to capture the evolution of grain orientation, the elastic rotation tensor $\mathbf{R}^e$ should be obtained. To do so, the elastic deformation gradient $\mathbf{F}^e$ can be decomposed as follows:

$$\mathbf{F}^e = \mathbf{R}^e \mathbf{U}^e \qquad (11)$$

where $\mathbf{U}^e$ is the elastic material stretch tensor. Calculation of $\mathbf{R}^e$ from Eq. (11) can then become an eigenvalue problem [39]. The elastic asymmetric spin tensor $\boldsymbol{\omega}^e$ can be obtained from $\mathbf{R}^e$ as follows:

$$\boldsymbol{\omega}^e = \dot{\mathbf{R}}^e \mathbf{R}^{eT}. \qquad (12)$$

The orientation of each grain can then be updated by incrementally integrating Eq. (12) for each simulation increment.

The homogenized Cauchy stress tensor of the sample $\boldsymbol{\sigma}^H$ can be obtained by averaging of the Cauchy stress tensors for each grain as follows:

$$\boldsymbol{\sigma}^H = \frac{1}{W} \sum_{k=1}^{I} \omega^k \boldsymbol{\sigma}^k \qquad (13)$$

where ω^k and $\boldsymbol{\sigma}^k$ are the kth grain volume and Cauchy stress tensor, respectively, I is the total number of grains, and W is the total sample volume which can be defined as $W = \sum_{k=1}^{I} \omega^k$.

The final aspect of the model is to capture grain reorientation due to twinning which is especially important in hcp metals and alloys. In the current framework, the PTR scheme developed by Tomé et al. [35] is incorporated to capture twinning. The twinning is modeled as pseudo-slip system, and the nominal twin volume fraction for k^{th} twin system, i.e., f_{tw}^{k*}, can be evolved as follows:

$$\dot{f}_{tw}^{k*} = \frac{\dot{\gamma}^k}{S} \qquad (14)$$

where S is the characteristic twin shear strain. According to the PTR scheme, when f_{tw}^{k*} for any twin system becomes larger than a specific threshold of F_T, the grain will be reoriented according to that twin system. The threshold F_T can be defined as follows [15, 35]:

$$F_T = A + B \frac{F_E}{F_R} \qquad (15)$$

where F_E is the total reoriented twin volume within the sample and F_R is the total nominal twin volume fraction within the sample.

After the grain is reoriented due to twinning, one cannot use the converged elastic and plastic deformation gradient tensor for the last increment as the starting values. This is because those computed state variables will not be consistent with the reoriented grain orientation. Accordingly, right after the reorientation, both plastic and elastic deformation gradient tensors are reset to the identity matrix, and the total deformation history is applied on that grain to reach a new state of strain and stress which is consistent with the new grain orientation.

It is worth mentioning that all the constitutive model calculations are performed within the sample frame. Accordingly, at the end of constitutive model calculations, no more transformation is required since we are already within the sample frame. This requires the transformation of the elastic stiffness tensor and Schmid tensor from the crystal frame to the sample frame at the start of each increment using the grain orientations.

Software Structure

The new rapid texture evolution analysis PRISMS-Plasticity TM pipeline is integrated as a part of the PRISMS-Plasticity software. The inputs include the microstructure, material models, and applied boundary conditions, while the outputs include grain-level stress and strain tensors, slip activities, and updated orientations.

The microstructure of the sample is the first input. This includes the list of orientations for each grain using Rodrigues vector representation. Since the PRISMS-Plasticity framework is used here, a grain ID list should be generated as well. The Taylor-type simulation does not, however, include any grain neighborhoods and thus any form of grain ID list is acceptable. The Dream.3D pipeline is available within the package and can be used to generate the required starting microstructure from input EBSD microstructures or synthetic microstructures.

All the material constants should be provided in the input file. This includes the number of slip/twin systems, initial slip resistances, isotropic and kinematic hardening parameters, latent hardening coefficients, crystallographic information of slip/twin normal and directions, reference shearing rate and inverse strain rate sensitivity exponent,

characteristic twin shear strain, twinning threshold parameters, and numerical integration parameters of the crystal plasticity model. The final input data are the applied boundary conditions. The boundary condition is defined here as an applied velocity gradient tensor. One should define the total simulation time in which the deformation gradient is applied.

The output includes all the grain-level information including stress and strain tensors, slip activity, twin volume, and updated orientations. How frequently these outputs are to be written can be set by the user. Using the grain-level outputs along with the corresponding grain sizes, one can extract the homogenized values for stress and strain tensors, slip activity, and twin volume. Also, the outputs can be read using a MATLAB script within the package which uses the MTEX toolbox to report the updated texture by plotting the updated pole figures and inverse pole figures.

The final step links the simulation inputs and outputs to Materials Commons as a virtual collaboration space and repository. Data can be transferred into Materials Commons in three different ways: (1) uploading using the website GUI, which is suitable for a small file or projects; (2) Using Globus which is especially helpful for a large dataset; or (3) Using Materials Commons Python Application Programming Interfaces (MCAPI) wrapper (*prismscpfe_mcapi*) which automates the integration of texture evolution results. *prismscpfe_mcapi* automatically uploads the simulation inputs and outputs into Materials Commons.

Application Examples

Three different examples are solved here to demonstrate the capabilities of this texture evolution framework. The first example is the evolution of texture during uniaxial compression of a polycrystalline OFHC copper sample with initial random texture. The importance of twinning on the texture evolution of Mg alloys is highlighted in the second example, in which the evolution of texture in extruded Mg alloy ZK60A sample is captured during the uniaxial compression along the extrusion direction. In the third example, the cyclic response of 7075-T6 Al is captured using the implemented back stress formulation. In the following examples, the reported true strain tensor $\boldsymbol{\varepsilon}_{\text{true}}$ is obtained as follows:

$$\boldsymbol{\varepsilon}_{\text{true}} = \int \mathbf{d}_{\text{BCs}} dt \xrightarrow{\text{constant} \mathbf{d}_{\text{BCs}}} \boldsymbol{\varepsilon}_{\text{true}} = \mathbf{d}_{\text{BCs}} t \tag{16}$$

where t is the simulation time and $\mathbf{d}_{\text{BCs}}$ is the rate of the deformation tensor applied as the boundary conditions as the BCs and can be obtained as follows:

$$\mathbf{d}_{\text{BCs}} = \frac{1}{2} \left(\mathbf{L}_{\text{BCs}} + \mathbf{L}_{\text{BCs}}^{\text{T}} \right) \tag{17}$$

where $\mathbf{L}_{\text{BCs}}$ is the velocity gradient tensor applied to the sample.

Polycrystalline OFHC Copper

The first example is to simulate the texture evolution of polycrystalline OFHC copper with random texture during uniaxial compression. The crystal plasticity parameters are calibrated using as a reference the experimental stress–strain response of the OFHC copper sample during uniaxial compression reported by Bronkhorst et al. [40]. Also, the PRISMS-Plasticity TM predicted response is also compared versus the simulation of Anand and Kothari [37] for the same material. The sample initially has a random texture consisting of 400 grains with equal sizes. The rate-dependent crystal plasticity formulation is used here with isotropic hardening, and no kinematic hardening is incorporated for this example, i.e., $h_1 = h_2 = r_1 = r_2 = m_1 = m_2 = 0$. To mimic the uniaxial compression along the z-axis, the velocity gradient tensor of $\mathbf{L}_{\text{compression}} = \begin{bmatrix} 0.0005 & 0 & 0 \\ 0 & 0.0005 & 0 \\ 0 & 0 & -0.001 \end{bmatrix} s^{-1}$ is incorporated. The elastic constants of $C_{11} = 170 \text{GPa}, C_{12} = 124 \text{GPa}$, and $C_{44} = 75 \text{GPa}$ are incorporated [37]. The crystal plasticity parameters calibrated by Anand and Kothari [37] are incorporated here as the starting point, which are $h_0^{\vartheta} = h_0 = 180 \text{MPa}, s_0^{\vartheta} = s_0 = 16 \text{MPa}, s_s^{\vartheta} = s_s = 148 \text{MPa}$, and $a^{\vartheta} = a = 2.25$ for all slip systems as described in Eq. (7). The latent hardening parameter of $q = 1.4$ and reference shearing rate of $\dot{\gamma}_0 = 0.001 s^{-1}$ are used as recommended in [37]. Finally, the inverse strain rate sensitivity exponent of $m = 75$ is selected for the simulation. The predicted response by Anand and Kothari [37] overpredicts the strength of sample at high strains. Accordingly, the crystal plasticity parameters are further calibrated to better capture the stress–strain response. The calibrated crystal plasticity parameters are $h_0^{\vartheta} = h_0 = 200 \text{MPa}$, $s_0^{\vartheta} = s_0 = 16 \text{MPa}$, $s_s^{\vartheta} = s_s = 129.5 \text{MPa}$, and $a^{\vartheta} = a = 2$. Figure 1 compares the PRISMS-Plasticity TM simulation results of OFHC copper sample during uniaxial compression versus the experimental results of Bronkhorst et al. [40] and simulation results of Anand and Kothari [37]. The results show that the PRISMS-Plasticity TM results agree well with the experimental results of Bronkhorst et al. [40]. The simulation takes 6.5 min for a single processor on PRISMS Lighthouse cluster. The PRISMS-Plasticity TM simulation prediction of the sample texture at the true
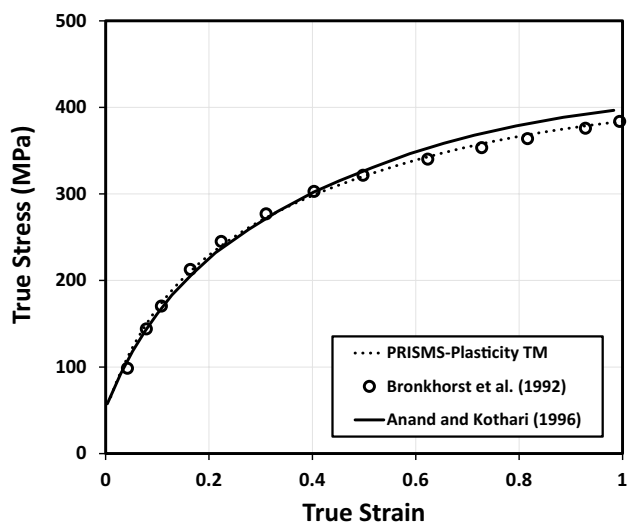
Fig. 1 The predicted stress–strain response of a polycrystalline OFHC copper sample during uniaxial compression test obtained by PRISMS-Plasticity TM. The results are compared with the experimental results of Bronkhorst et al. [40] and crystal plasticity simulation of Anand and Kothari [37]

strains of 99% compressive strain is compared versus the experimental results of Bronkhorst et al. [40], as shown in Fig. 2. The simulated texture reported by Anand and Kothari [37] for the true strain of 100% is also presented in Fig. 2. The characteristics of the predicted textures agree well with the results presented by Anand and Kothari [37] and Bronkhorst et al. [40].

A scaling study was conducted to investigate the parallel performance of the PRISMS-Plasticity TM pipeline. In the case of strong-scaling, different processor numbers of 2, 4, 16, and 64 are used to simulate the OFHC copper sample. For the scaling study, the assumption of perfect plasticity is incorporated. Figure 3b shows that the software scales nearly ideally for the number of processors evaluated. A slight performance drop for 64 processors can be attributed to the fact that the ratio of (number of grains/number of processors) is small for 64 processors. Using 64 processors this simulation took approximately 8 s to complete. To investigate the weak-scaling, the required simulation times for different sample sizes but a similar ratio of (number of grains/number of processors) = 400 are presented in Fig. 4. The results show almost ideal weak-scaling up to the sample with 25,600 grains simulated using 64 processors. However, the scaling slightly drops for the sample of 102,400 grains simulated using 256 processors. Compared to the CPFE software PRISMS-Plasticity [15], both weak-scaling and strong-scaling are significantly enhanced for the PRISMS-Plasticity TM texture evolution pipeline.

ZK60A Mg Alloy

Extruded Mg alloys frequently exhibit a texture with basal poles aligned perpendicular to the extrusion direction [15, 24, 41]. In materials with this texture, when loaded in compression along the extrusion direction, extension twinning plays a key role in sustaining deformation along the hcp c-axis. The PRISMS-Plasticity TM with rate-dependent crystal plasticity formulation and isotropic hardening is incorporated to model the response of extruded ZK60A Mg alloy during uniaxial compression and tension along the extrusion direction. No kinematic hardening is incorporated for this example, i.e., $h_1 = h_2 = r_1 = r_2 = m_1 = m_2 = 0$. The sample consists of 1080 grains with equal sizes. The assumed initial texture of the sample is presented in Fig. 5, which shows that the majority of grains have their basal poles aligned perpendicular to the extrusion direction. The uniaxial compression and tension along the extrusion direction (z-axis) are modeled using the velocity gradient tensor of $\mathbf{L}_{\text{compression}} = \begin{bmatrix} 0.005 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & -0.01 \end{bmatrix} s^{-1}$ and $\mathbf{L}_{\text{tension}} = \begin{bmatrix} -0.005 & 0 & 0 \\ 0 & -0.005 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} s^{-1}$, respectively. Three slip modes of Basal $\langle a \rangle (\{0001\}\langle 11\bar{2}0\rangle)$, Prismatic $\langle a \rangle\left(\left\{10\bar{1}0\right\}\langle 11\bar{2}0\rangle\right)$, and Pyramidal $\langle c+a \rangle\left(\left\{\bar{1}\bar{1}22\right\}\langle\bar{1}\bar{1}23\rangle\right)$ are used along with an extension twin mode $\left(\left\{10\bar{1}2\right\}\langle\bar{1}011\rangle\right)$ to capture the response of ZK60A Mg alloy. The elastic constants of unalloyed Mg presented in Table 1 are incorporated here. Similar to Qiao et al. [42], the latent hardening parameter of $q = 1$ is used for the crystal plasticity model. The reference shearing rate of $\dot{\gamma}_0 = 0.001 s^{-1}$ and inverse strain rate sensitivity exponent of $m = 75$ are selected. The crystal plasticity parameters for ZK60A alloy are calibrated using the experimental results reported by Wu [43] for extruded Mg alloy ZK60A sample at room temperature during uniaxial compression and tension experiments. Table 2 presents the calibrated crystal plasticity parameters for slip and twin systems. The PTR parameters of $A = 0.45$ and $B = 0$ are used to capture the reorientation due to extension twinning. These constants are calibrated to accurately capture the length of the plateau and the steepness of the subsequent strain hardening during uniaxial compression test. Figure 6 compares the PRISMS-Plasticity TM simulation results of extruded ZK60A Mg alloy sample during uniaxial compression and tension along the extrusion directions versus the experimental results of Wu [43]. The PRISMS-Plasticity TM successfully captures the responses of sample during both uniaxial tension and compression loadings. In order to investigate the observed anisotropy in Fig. 5, the evolution of twin volume fraction is further investigated in Fig. 7. The results show that the

Fig. 2 The final crystallographic texture of polycrystalline OFHC copper with random initial texture: **a** experimental texture after a true strain of − 99% reported by Bronkhorst et al. [39], **b** simulated texture after a true strain of − 100% reported by Anand and Kothari [40] and **c** PRISMS-Plasticity TM simulated texture after a true strain of–99% (after Bronkhorst et al. [39] and Anand and Kothari [40])
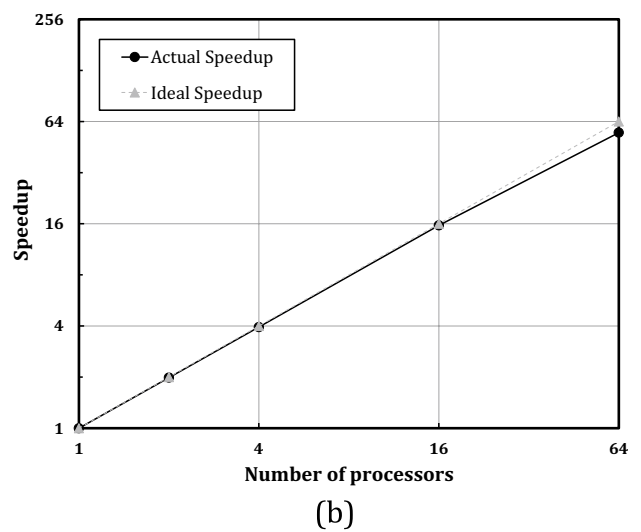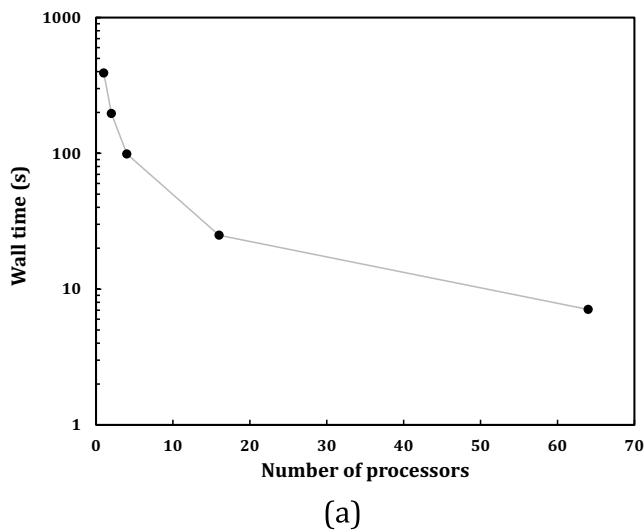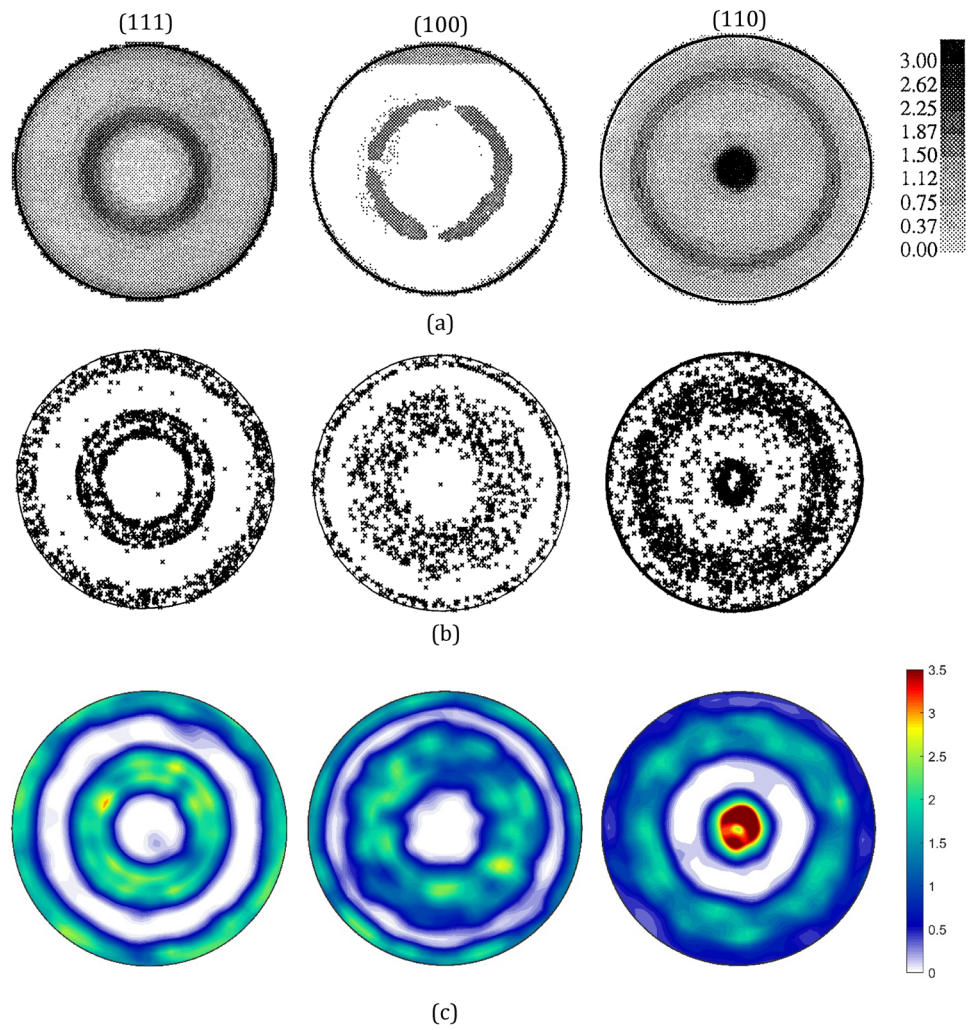
(111)　　　　(100)　　　　(110)

(a)

(b)

(c)

Fig. 3 Strong-scaling computational performance for the PRISMS-Plasticity TM rapid texture evolution analysis pipeline: **a** simulation wall time versus the number of processors and **b** actual speedup versus the ideal speedup for different numbers of processors. A polycrys-talline copper sample containing 400 grains with random initial texture subjecting to 100% true compressive strain is investigated in the strong-scaling study
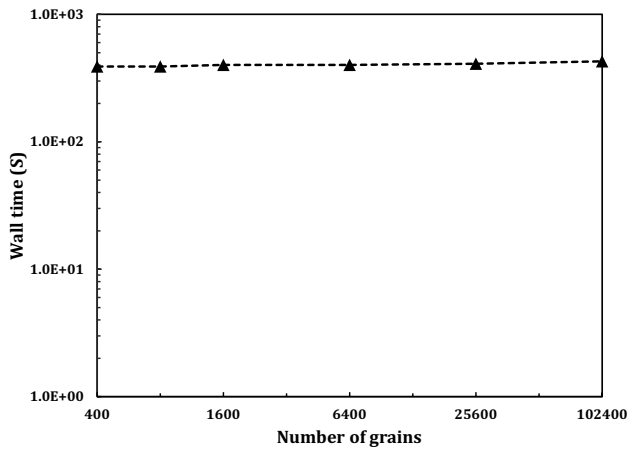
Fig. 4 Weak-scaling computational performance for the rapid texture evolution analysis pipeline. A polycrystalline copper sample with random initial texture subjected to 100% true compressive strain is investigated in the weak-scaling study. The ratio of (number of grains/number of processors) = 400 is initially selected, and the performance is investigated as the number of grains is increased to 102,400

observed anisotropy in Fig. 5 can be attributed to the activation of twinning during compression loading along the extrusion direction while the twin content is negligible during the uniaxial tension. The compressive and tensile simulations take 6.6 min and 2.4 min using 64 processors on the Stampede2 cluster, respectively.

In the final step, the evolution of basal (0001) pole figures of extruded ZK60A Mg alloy during the compression loading along the extrusion direction is investigated, as shown in Fig. 8. The twinning deformation mechanism leads to an 86.3° reorientation of the basal pole which increases the basal {0002} peak intensity along the extrusion direction. Comparing Fig. 8 with Fig. 7, one can see that the larger twin volume fractions correlate with the higher basal {0002} peak intensity in the center of the pole figures shown in Fig. 8. Also, one can see that the grains with basal poles aligned closer to the normal to the loading direction reorient due to twinning at lower strains compared to the ones with poles deviating further away from normal to loading direction. Almost 73% of the grains are reoriented due to
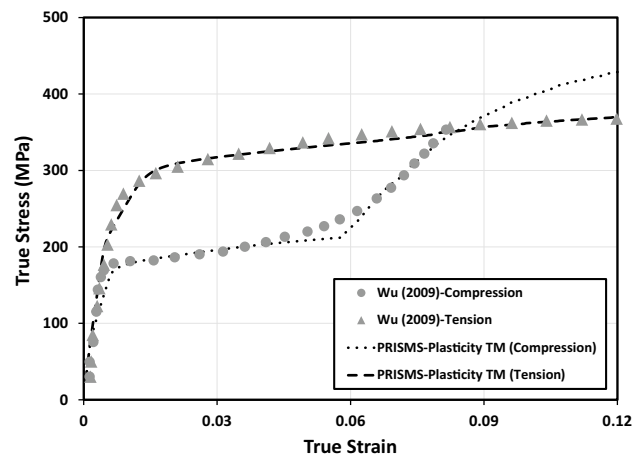
Fig. 5 The initial texture of the extruded ZK60A Mg alloy sample. The loading direction is along the extrusion direction, which is out of plane of the image. The majority of grains have their basal poles aligned perpendicular to the extrusion direction
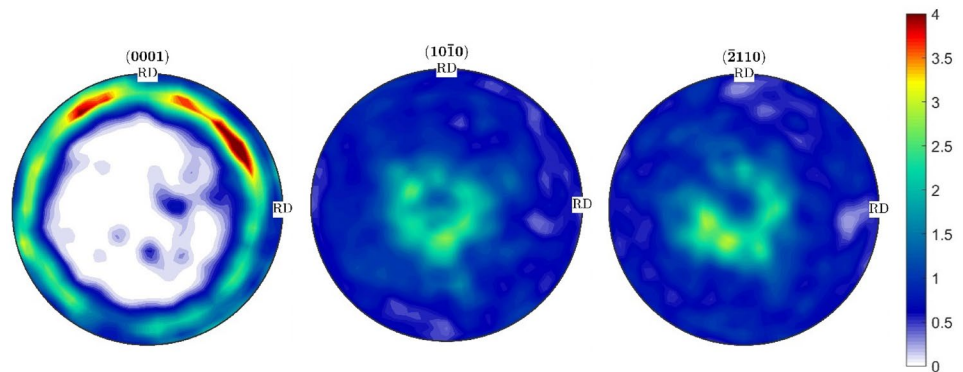
Table 1 The elastic constants (MPa) of Mg at room temperature [50]

C11	C12	C13	C33	C44
59,400	25,610	21,440	61,600	16,400

Table 2 Crystal plasticity parameters of Mg alloy ZK60A for different slip and twin systems

Mode	s_0^α(MPa)	h_0^α(MPa)	s_s^α(Mpa)	θ^α
Basal	20	10	35	2
Prismatic	100	800	155	2
Pyramidal <c + a>	207	1200	340	2
Twinning	36	400	40	0

Fig. 6 The stress–strain response of the extruded ZK60A Mg alloy sample during uniaxial compression and tension test along the extrusion direction obtained by PRISMS-Plasticity TM. The results are compared with the experimental results of Wu [43]
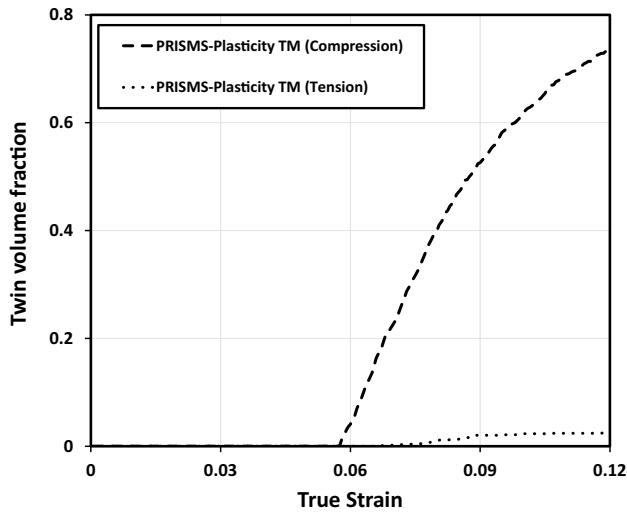
twinning at the compressive strain of 12% along the extrusion direction.

The current version of the PRISMS-Plasticity TM texture analysis pipeline can capture twinning which is suitable for modeling of monotonic loading. However, during more complex loading paths such as cyclic experiments, detwinning may occur as well. In a future version, a similar scheme to the twinning–detwinning framework developed by Wang et al. [44] will be implemented for the texture analysis pipeline to capture both twinning and detwinning mechanisms. A twinning–detwinning model based on this concept has been implemented in the CPFE model in PRISMS-Plasticity [24].

7075-T6 Aluminum Alloy

The third example is to capture the response of the aluminum alloy 7075-T6 during cyclic loadings. Unlike the previous two examples where the monotonic loading is simulated using the isotropic hardening, the cyclic response of the sample cannot be handled using the isotropic hardening, and the back stress formulation is required to capture the cyclic response. Accordingly, the rate-dependent crystal plasticity model with a two-term back stress Ohno–Wang type is used as described in Eq. (8). The sample initially has a random texture consisting of 400 grains with equal sizes. To mimic the compressive and tensile part of the cyclic response along the z-axis, the velocity gradient tensors of

$$\mathbf{L}_{\text{compression}} = \begin{bmatrix} 0.005 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & -0.01 \end{bmatrix} s^{-1} \quad \text{and} \quad \mathbf{L}_{\text{tension}} = \begin{bmatrix} -0.005 & 0 & 0 \\ 0 & -0.005 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} s^{-1} \text{ are incorporated, respectively.}$$

The crystal plasticity parameters are calibrated versus the experimental response of the 7075-T6 Al sample during fully reversed strain controlled cyclic loading with different strain amplitudes reported by Arcari [45]. The elastic constants of $C_{11} = 107.3$GPa, $C_{12} = 60.9$GPa, and $C_{44} = 28.3$GPa are incorporated, as reported by Bozek et al. [46]. The reference shearing rate of $\dot{\gamma}_0 = 0.001s^{-1}$ and inverse strain rate sensitivity

Table 3 Crystal plasticity parameters of 7075-T6 Al

s_0^α(MPa)	h_1(MPa)	r_1	h_2(MPa)	r_2	$m_1 = m_2$
35	1.5×10^5	2500	7.4×10^4	710	70

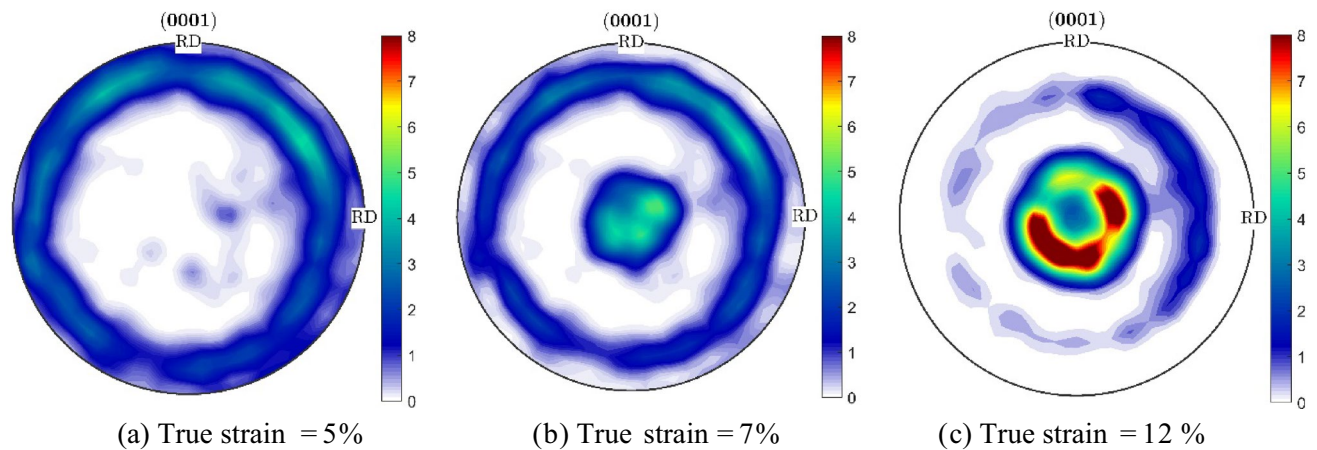(a) True strain = 5%   (b) True strain = 7%   (c) True strain = 12 %

Fig. 8 The PRISMS-Plasticity TM simulated evolution of basal (0001) pole figures of extruded ZK60A during the compression loading along the extrusion direction. Note: The twinning deformation mechanism leads to an 86.3° reorientation of the basal pole which increases the basal {0002} peak intensity along the extrusion direction
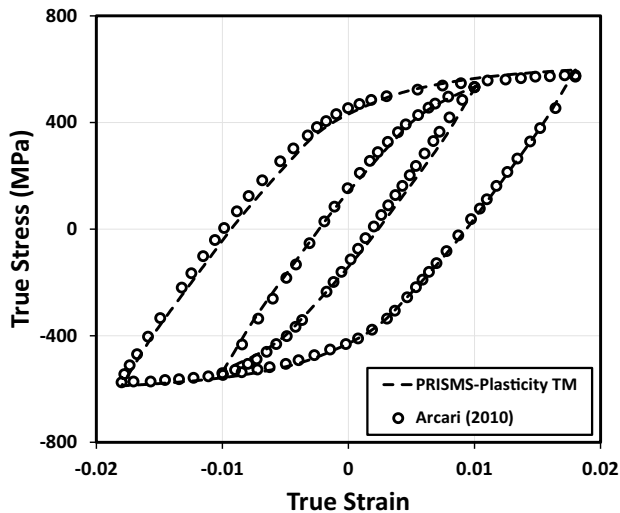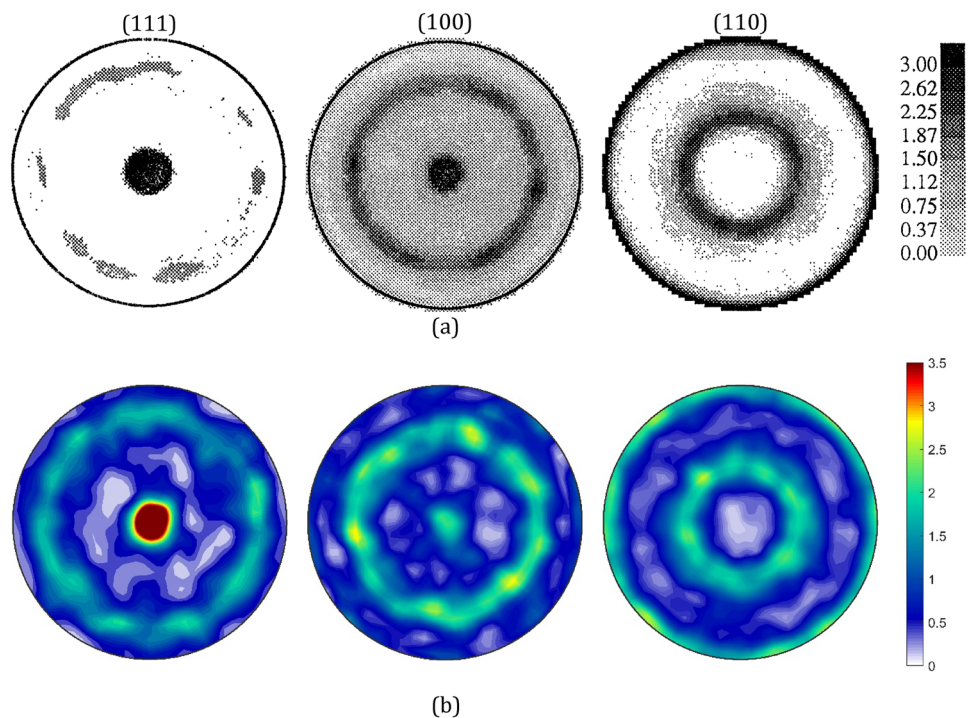
Fig. 9 Cyclic response of 7075-T6 Al subjected to strain amplitudes of 1% and 1.8% predicted by PRISMS-Plasticity TM. The results are compared with the experimental results of Arcari [45]

exponent of $m = 75$ are selected. Table 3 presents the calibrated crystal plasticity parameters for cyclic response of 7075-T6 Al. Figure 9 compares the cyclic response of the 7075-T6 Al sample subjected to strain amplitudes of 1% and 1.8% predicted by PRISMS-Plasticity TM versus the experimental results of Arcari [45]. The results show that PRISMS-Plasticity TM can accurately capture the cyclic response of the 7075-T6 Al sample at different strain amplitudes. The cyclic simulations with strain amplitudes of 1%

and 1.8% require 3 min and 4.8 min using a single processor on the Stampede2 cluster, respectively. In the last step, the texture evolution of the 7075-T6 Al sample at the uniaxial tensile strain of 37% is predicted using the calibrated PRISMS-Plasticity TM framework, as shown in Fig. 10. Texture results for this precise condition are not available for fcc aluminum alloy 7075-T6, therefore they are compared with the results of fcc copper described in Sect. 4.1. The results are compared versus the texture of OFHC copper sample at the same tensile strain as reported by Bronkhorst et al. [40]. The characteristics of the predicted textures agree well with the experimental results reported by Bronkhorst et al. [40].

Application to Machine Learning Frameworks

The PRISMS-Plasticity TM pipeline is specifically designed to for high-throughput simulations. An important application for this is generation of large datasets for use with machine learning. In this section, we describe one such application for manufacturing process optimization of 7075-T6 Al. A simulated sample of this alloy is subjected to mechanical deformation sequences comprising up to three stages. A rate-dependent crystal plasticity formulation is used. The elastic constants of $C_{11} = 107.3\text{GPa}$, $C_{12} = 60.9\text{GPa}$, and $C_{44} = 28.3\text{GPa}$ are incorporated [29, 30]. An initial slip resistance of 35 MPa is selected for all 12 slip systems [29, 30] with

Fig. 10 The crystallographic texture of fcc polycrystalline metals with initial random texture during the uniaxial tension after a true strain of 37%: **a** experimental results of polycrystalline OFHC copper reported by Bronkhorst et al. [40] and **b** PRISMS-Plasticity TM simulation of 7075-T6 Al (after Bronkhorst et al. [39])

Table 4 The velocity gradient tensor L for six processes used in the machine learning application. In the rolling process, the second direction is the rolling direction while the first one is the compression direction. For example, in "Rolling *YX*" process, *X* is the rolling direction while *Y* is the compression direction

Tension *X*	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}$
Tension *Y*	$\begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}$
Tension *Z*	$\begin{bmatrix} -0.5 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Rolling *YX*	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Rolling *ZY*	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$
Rolling *XZ*	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

no hardening. A variational autoencoder (VAE) network is used in the current work to obtain a continuous latent space for the generated database [20]. The PRISMS-Plasticity TM pipeline is used to develop the training and validation database. For each step, the applied velocity gradient tensor **L** is selected from six processes (Table 4), each is multiplied by four different multipliers of $\alpha = \{-1, -0.5, 0.5, \text{and} 1\}$, i.e., 24 different possibilities are available for each process step. Each process step is modeled for $0.2s$. In total, 14,424 process sequences are modeled using the PRISMS-Plasticity TM framework. The sample initially has a random texture consisting of 8000 grains with equal sizes. The result of each simulation is a discretized orientation distribution function (ODF). Out of the resulting 14,424 ODF pairs, 10,424 pairs are incorporated for training, 2000 pairs are used for validation, and 2000 data points are used as test sets.

A variational autoencoder (VAE) is incorporated here, which is a class of autoencoders that is generative [47, 48]. VAE consists of an encoder, latent space, and a decoder. The encoder reduces the dimensionality of the input data into a constrained multivariate latent distribution, while the decoder reconstructs the output data from the low-dimension latent space. To train the VAE, ODFs from training sets are fed to the encoder. The encoder compresses input ODFs into lower dimension latent space, which has user-defined dimensionality. The decoder can convert the lower-dimensional data from the latent space into the ODF output. In order to train the VAE model, an error parameter is established which consists of two terms. The first error term is the reconstruction loss which is the difference between the input ODFs and output ODFs for training sets. The second

term is Kullback–Leibler (KL) divergence which defines how the trained latent distribution deviates from the known prior distribution of the latent space, which is commonly a spherical normal distribution [47, 48]. The Adam optimization algorithm is used to train the model. The model is trained by minimizing the defined error. Next, the trained model is evaluated using the validation datasets to ensure that the trained model is not overfitted. Keras, which is a deep learning API written in Python, is used in the current work to set the VAE model [49].

In the first step, the input data is trained for 1-D−5-D latent spaces. A n-D latent space presents the data in an n-dimensional space. In the case of 2-D latent space, all the data can be visualized using a 2-D plot, as shown in Fig. 11. In these three plots, different colors have been used to show the intensity of the yield stress and elastic modulus along the x direction, i.e., σ_x and E_x, respectively, for various process sequences defined in 2-D latent space. As can be seen in Fig. 11a, the elastic modulus is maximized for Z1-Z2 pairs that lie on the outer circumference of the plot. A similar behavior can be observed for yield strength in Fig. 11b. One can also define the objective to maximize the yield strength while minimizing the elastic modulus. To do so, the ratio of the yield strength over the elastic modulus along the x direction (σ_x / E_x) is also used in latent space 2-D plot as shown in Fig. 11c. In this case the maximum values of this ratio for values of Z1-Z2 pairs that closer to the center of the dataset on the 2-D latent space plot. Within the range of the applied processes, the final elastic modulus along the loading axes is within the range of 69.5–73 GPa while the final yield stress for loading along x-axis is within the range of 106–109.5 MPa. The simulation time of 0.2 s is selected for each stage. One can either increase the simulation time to apply larger strains or increase the number of stages for each process to observe wider spread for final elastic modulus and yield stress.

The validation sets are used to evaluate the trained models. Figure 12a shows the normalized L_2 reconstruction loss for different latent spaces obtained for validation sets. The errors are higher for 1-D latent spaces compared to higher-dimensional latent spaces. The results show that using 2-D latent space would be optimal for the three-stage processes space. The error observed in Fig. 12a is mainly due to the loss in the intensity in the output ODF. This is attributed to the scaling of input data and descaling of the output. This is required to keep all attributes of the input data between 0 and 1, which is one of the assumptions of the methods used, and the NN layers. The scaling is required by the VAE technique since all the inputs for the encoder should be between 0 and 1. However, the trained VAE models conserve the primary fibers of the input ODF, as shown in Fig. 12b for three randomly selected input ODFs. Similar to Fig. 12a, Fig. 12b shows that all latent spaces with dimensions higher

Fig. 11 Representation of ODFs generated by PRISMS-Plasticity TM in 2-D latent space for training sets. Each process sequence is represented by a single point in latent space and colored based on its: **a** E_x, **b** σ_x, and **c** σ_x/E_x. Z1-Z2 represents the 2-D latent space
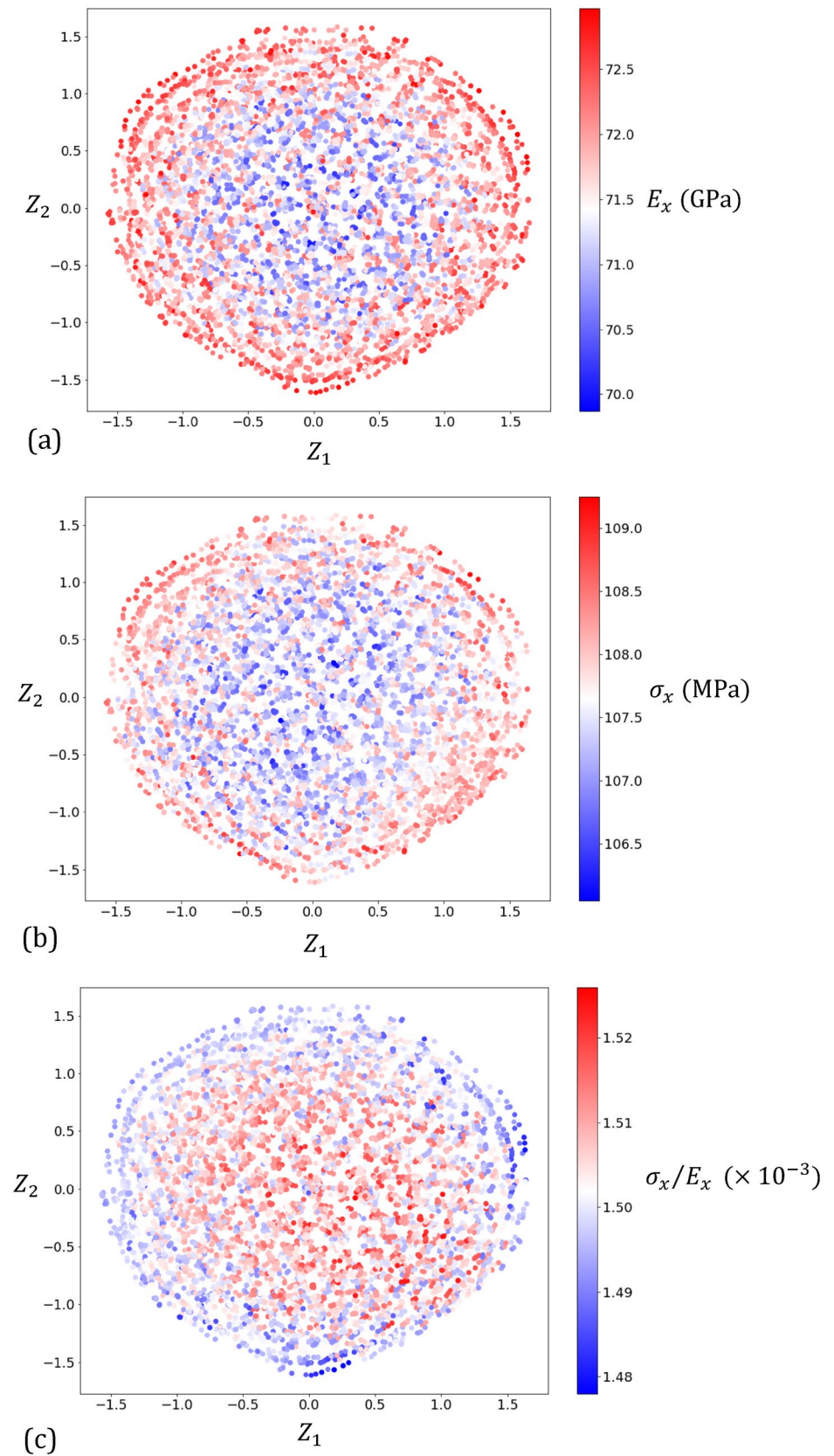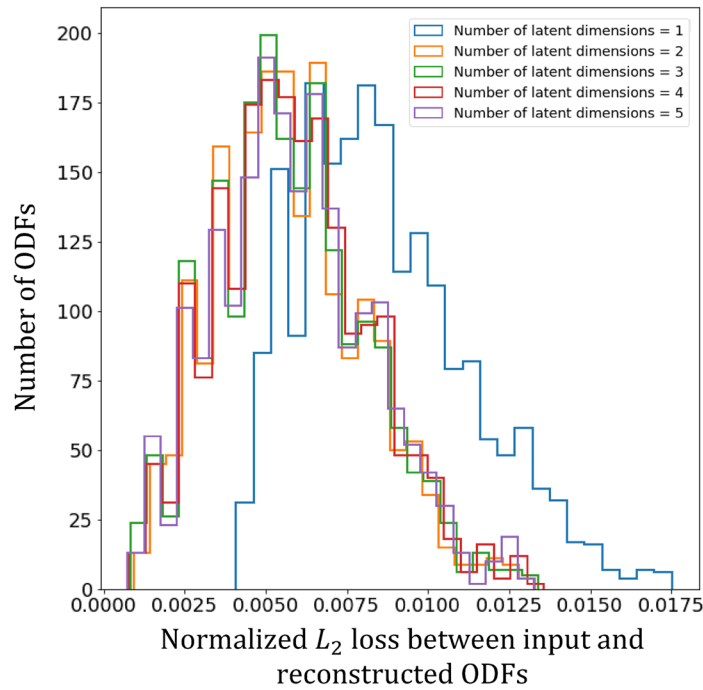
(a)
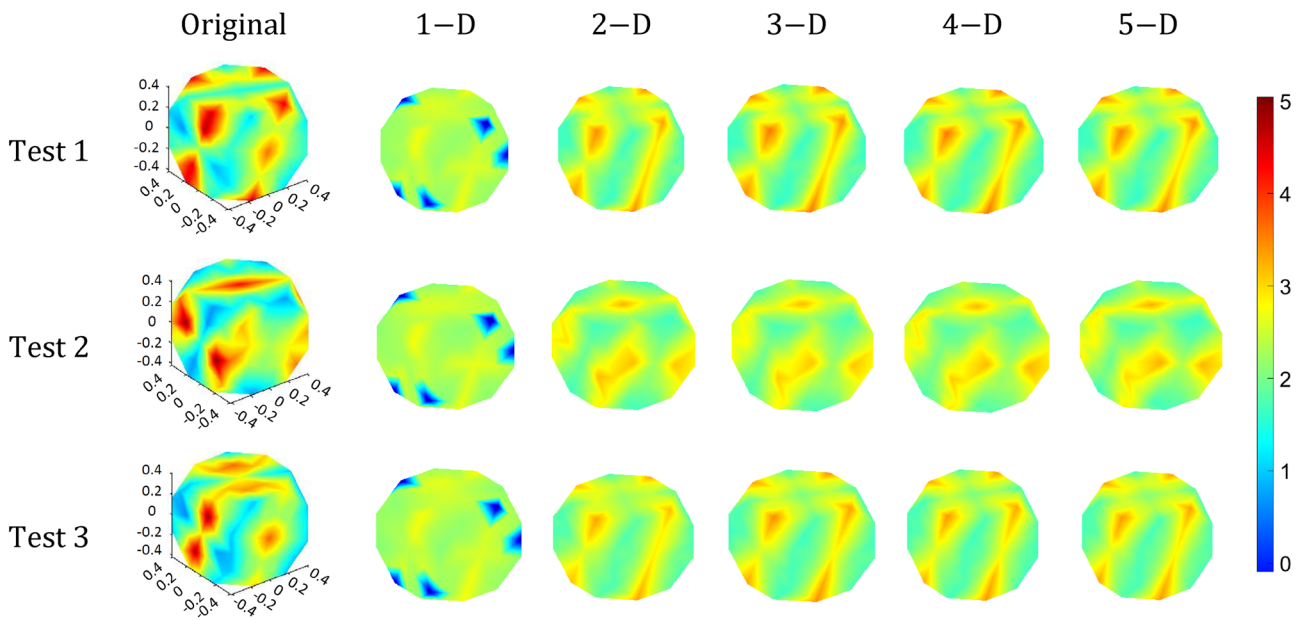
(b)

Fig. 12 Evaluation of the trained VAE model using the PRISMS-Plasticity TM generated validation sets: **a** quantitative analysis of the trained VAE model using the normalized reconstruction loss for different latent spaces and **b** qualitative analysis of the trained VAE models by comparing three randomly selected input ODFs and the corresponding reconstruction ODFs. The ODFs are plotted in the Rodrigues fundamental region, and the coordinate system is presented for the original ODFs, which is the same for the reconstructed ones

than 1-D replicate the important fibers of the input ODF. The smallest latent space that can have acceptable reconstruction error is the optimum space. The results show that the 2-D latent space is the optimum representation of the ODF map for three-stage processes.

The last step is to illustrate sequence identification. To do so, two different input ODFs are selected from the test sets. Surrounding each of these two points in the latent space, there is a cluster of ODFs predicted by the trained VAE. Figure 13 shows the normalized L2 norm of the difference between the input ODF and the predicted ones normalized by the L2 norm of the input ODF. Some of the predicted ODFs are represented as well, which are highlighted by red points on the curve. The predicted ODFs for the sequences of A1 and A2 are very close to the ODFs of the input ODFs for Cases 1 and 2, respectively. Figure 13 shows that as the reconstruction error increases, the corresponding predicted ODFs deviate from the ODF of the corresponding input cases. The process paths for the two input ODFs and the predicted ODFs plotted for each of those are presented in Tables 5 and 6, respectively. The ODF for the predicted sequences of A1 and B1 has the least difference with the input ODF for Case 1 among all the plotted ODFs. It is interesting that one can mimic the three-stage process of the input process of Case 1 with just the two-step process of B1. In the case of the input ODF for Case 2, the closest predicted ODF is A2 which has different processes compared to the input process for Case 2 but leads to a similar ODF. Thus, one can explore more economical processing routes using this database to achieve target textures.

One can use the sequence identification process for manufacturing process design to reach certain properties with alternatives (and potentially simpler) process steps. The relatively simple example of the ratio of yield strength over the elastic modulus along x direction, i.e., σ_x/E_x, is used here

Fig. 13 Sequence identification using VAE model. Two different input ODFs shown on the left of the figure are selected from the test sets. Surrounding each of these two points in the latent space (Fig. 11), there are a cluster of ODFs predicted by the trained VAE. The normalized L2 norm of the difference between the input and predicted ODFs is presented here. Some of the predicted ODFs are represented as well, which are highlighted by red points on the curve. The ODFs are plotted in the Rodrigues fundamental region
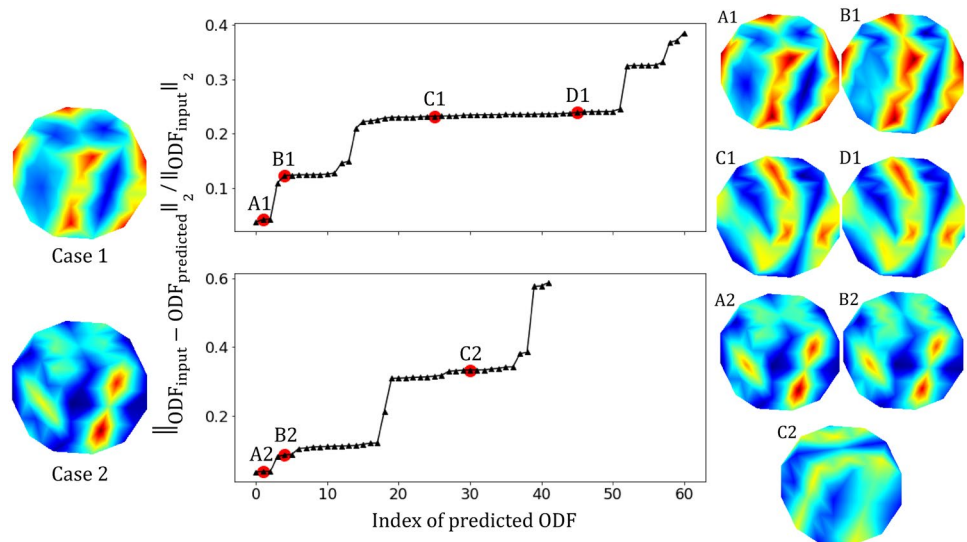
Table 5 Process sequences for the input cases of Fig. 13

	Stage 1	α_1	Stage 2	α_2	Stage 3	α_3
Case 1	Rolling YX	−0.5	Tension X	1.0	Tension Z	1.0
Case 2	Tension Z	−1.0	Rolling YX	0.5	Rolling XZ	0.5

Table 6 Process sequences predicted for the input cases of Fig. 13

	Stage 1	α_1	Stage 2	α_2	Stage 3	α_3
A1	Tension Z	1.0	Rolling YX	1.0	Tension X	1.0
B1	Rolling XZ	−0.5	Tension Z	1.0	NaN	NaN
C1	Rolling ZY	−0.5	Tension Z	0.5	Tension Z	−0.5
D1	Rolling XZ	1.0	Rolling XZ	−1.0	Rolling ZY	−0.5
A2	Rolling XZ	0.5	Rolling YX	0.5	Tension Z	−1.0
B2	Tension Y	−1.0	Rolling ZY	0.5	Rolling ZY	0.5
C2	Tension Z	−1.0	Tension Z	1.0	Rolling XZ	−0.5

Fig. 14 Sequence identification using the VAE model. Starting ODFs are on left of figure and same as in Fig. 13. ODF identification is obtained by matching the ratio of yield strength over the elastic modulus along the x direction, i.e., σ_x/E_x, between the input and predicted ODFs. The ODFs are plotted in the Rodrigues fundamental region
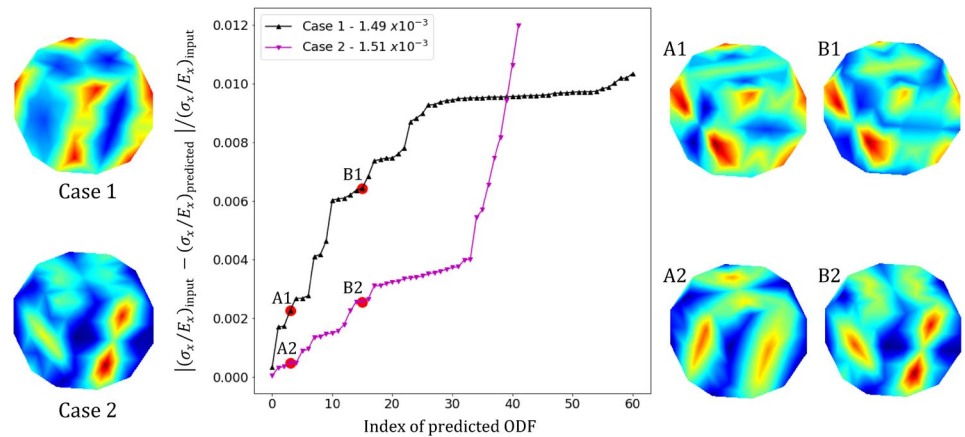

Table 7 Process sequences predicted for the input cases of Fig. 14

	Stage 1	α_1	Stage 2	α_2	Stage 3	α_3
A1	Rolling XZ	0.5	Tension X	1.0	Rolling XZ	0.5
B1	Tension Z	1.0	Tension Y	−1.0	Rolling ZY	1.0
A2	Rolling ZY	−0.5	Tension Z	−1.0	NaN	NaN
B2	Tension Y	−1.0	Rolling ZY	0.5	Rolling ZY	0.5

as the design property. The goal is to achieve the same ratio of σ_x/E_x but with fewer or simpler process steps. Figure 14 shows the difference between the σ_x/E_x ratio for the input ODFs and the predicted ones. Two predicted ODFs with σ_x/E_x almost equal to each input ODF are also presented which correspond to the red points on the curve. Although the values of σ_x/E_x are very close to those of the input ODFs, the predicted ODFs are different compared to the input ones both in intensity and general fibers. Table 7 shows the process sequences related to each set of predicted ODFs presented in Fig. 14. Again, one can select different combinations of process sequences that result in similar property ratio σ_x/E_x. In Case 2, a process sequence containing only two steps (Sequence A2) was determined to have the same ratio as the input process rather than three steps used for the input process and the Sequence B2.

Summary and Conclusions

An efficient open-source texture modeling pipeline, PRISMS-Plasticity TM, is integrated into the CPFE software PRISMS-Plasticity [15] to rapidly capture the texture evolution of metallic samples subjected to mechanical processes. PRISMS-Plasticity TM uses a Taylor-type crystal plasticity model to overcome the limitations of CPFE that enforces solving the equilibrium equation over the discretized domain. Because of this limitation, CPFE codes are not efficient for global searches of different processes to optimize the specific response of metals and

alloys. In contrast, PRISMS-Plasticity TM relies on the extensive texture and microstructural mapping pipelines established for PRISMS-Plasticity along with its sophisticated library of materials model to capture the texture evolution using a Taylor-type model to minimize the cost of texture evolution analysis. The current framework can also easily reach very high strains, which is critical for many applications.

Both strong-scaling and weak-scaling of the developed texture evolution analysis pipeline are investigated and show nearly ideal scalability for the framework. Three examples of a polycrystalline OFHC copper sample with random texture during uniaxial compression, an extruded ZK60A Mg alloy subjected to the uniaxial compression, and a 7075-T6 aluminum alloy with random texture during cyclic loading and uniaxial tension are investigated. The results show that the PRISMS-Plasticity TM can successfully capture the main features of the texture development in the simulated examples. In addition to the slip deformation modes, the effect of extension twinning mode on texture evolution is incorporated in the case of ZK60A Mg alloy using the implemented PTR scheme.

The application of the PRISMS-Plasticity TM outputs to the machine learning framework is presented. The PRISMS-Plasticity TM pipeline is used to generate high-throughput data in 7075-T6 Al to explore three-stage process space for training and validating a variational autoencoder (VAE) machine learning model. The high-dimensional space of orientation distribution function is translated into the low-dimensional latent space, which can

be readily visualized and explored. The developed framework can be used for sequence identification to find different design paths for an optimized property. Here, the ratio of the yield strength normalized by the elastic modulus along the x-axis is selected as the target property set. The results show that the VAE model can successfully identify different process sequences which results in similar properties, allowing exploration of more economical process sequences to achieve desired textures.

Appendix 1

- PRISMS-Plasticity software can be freely downloaded from the following link: https://github.com/prisms-center/plasticity.
- prismscpfe_mcapi Python package can be freely downloaded from the following link: https://github.com/prisms-center/prismscpfe_mcapi.

Appendix 2

The PRISMS-Plasticity TM pipeline has a user-friendly input and output formats, which follows those of the PRISMS-Plasticity CPFE software. A user can add a flag in the main input file (prm.prm) to switch between the CPFE simulation and Taylor-type calculations. Here are the list of inputs and outputs for the framework:

Inputs

- prm.prm: The main input file for the simulation.
- grainID.txt: The list of grainIDs for each voxel. In the case of PRISMS-Plasticity Taylor model, each voxel represents a grain, and the order of the voxels does not influence the results.
- orientations.txt: The orientation of each grain which is defined as a Rodrigues–Frank vector.
- slipDirections.txt: The slip direction vector for each slip system.
- slipNormals.txt: The normal vector to slip plane for each slip system.
- twinDirections.txt: The pseudo-slip direction vector for each twin system.
- twinNormals.txt: The normal vector to twin plane for each twin system.
- LatentHardeningRatio.txt: The coefficient matrix which defines the hardening due to the interaction of different deformation modes.

Outputs

- stresstrain.txt: The average response of the sample including six components of stress and strain tensors along with the twin volume and plastic slip.
- QuadratureOutputs.csv: The simulation history for each grain. This includes the updated grain orientations, stress and strain tensors, and all the crystal plasticity state variables.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Taylor GI (1938) Plastic strain in metals. J Inst Metals 62:307–324
2. H. Honneff, H. Mecking (1981) Analysis of the deformation texture at different rolling conditions, p 347–355
3. Kocks UF, Chandra H (1982) Slip geometry in partially constrained deformation. Acta Metall 30(3):695–709
4. Van Houtte P (1982) On the equivalence of the relaxed Taylor theory and the Bishop-Hill theory for partially constrained plastic deformation of crystals. Mater Sci Eng 55(1):69–77
5. Van Houtte P, Delannay L, Samajdar I (1999) Quantitative prediction of cold rolling textures in low-carbon steel by means of the LAMEL model. Textures Microstruct 31(3):109–149
6. Van Houtte P, Delannay L, Kalidindi SR (2002) Comparison of two grain interaction models for polycrystal plasticity and deformation texture prediction. Int J Plast 18(3):359–377
7. Crumbach M, Goerdeler M, Gottstein G, Neumann L, Aretz H, Kopp R (2003) Through-process texture modelling of aluminium alloys. Modell Simul Mater Sci Eng 12(1):S1
8. Crumbach M, Goerdeler M, Gottstein G (2006) Modelling of recrystallisation textures in aluminium alloys: I. Model set-up and integration. Acta Mater 54(12):3275–3289
9. Tjahjanto DD, Eisenlohr P, Roters F (2009) A novel grain cluster-based homogenization scheme. Modell Simul Mater Sci Eng 18(1):015006
10. Lebensohn RA, Tomé CN (1993) A self-consistent anisotropic approach for the simulation of plastic deformation and texture

development of polycrystals: application to zirconium alloys. Acta Metall Mater 41(9):2611–2624

11. Turner PA, Tomé CN (1994) A study of residual stresses in Zircaloy-2 with rod texture. Acta Metall Mater 42(12):4143–4153

12. Wang H, Wu PD, Tomé CN, Huang Y (2010) A finite strain elastic–viscoplastic self-consistent model for polycrystalline materials. J Mech Phys Solids 58(4):594–612

13. Moulinec H, Suquet P (1998) A numerical method for computing the overall response of nonlinear composites with complex microstructure. Comput Methods Appl Mech Eng 157(1–2):69–94

14. Lebensohn RA (2001) N-site modeling of a 3D viscoplastic polycrystal using fast Fourier transform. Acta Mater 49(14):2723–2737

15. Yaghoobi M, Ganesan S, Sundar S, Lakshmanan A, Rudraraju S, Allison JE, Sundararaghavan V (2019) PRISMS-Plasticity: an open-source crystal plasticity finite element software. Comput Mater Sci 169:109078

16. Schmidt J, Marques MRG, Botti S, Marques MAL (2019) Recent advances and applications of machine learning in solid-state materials science. NPJ Comput Mater 5(1):83

17. Qi X, Chen G, Li Y, Cheng X, Li C (2019) Applying neural-network-based machine learning to additive manufacturing: current applications, challenges, and future perspectives. Engineering 5(4):721–729

18. Meng L, McWilliams B, Jarosinski W, Park H-Y, Jung Y-G, Lee J, Zhang J (2020) Machine learning in additive manufacturing: a review. JOM 72(6):2363–2377

19. Wang C, Tan XP, Tor SB, Lim CS (2020) Machine learning in additive manufacturing: state-of-the-art and perspectives. Addit Manuf 36:101538

20. Sundar S, Sundararaghavan V (2020) Database development and exploration of process–microstructure relationships using variational autoencoders. Mater Today Commun 25:101201

21. Bock FE, Aydin RC, Cyron CJ, Huber N, Kalidindi SR, Klusemann B (2019) A review of the application of machine learning and data mining approaches in continuum materials mechanics. Front Mater 6:110

22. Jinnouchi R, Karsai F, Kresse G (2020) Making free-energy calculations routine: combining first principles with machine learning. Phys Rev B 101(6):060201

23. Liu X, Athanasiou CE, Padture NP, Sheldon BW, Gao H (2020) A machine learning approach to fracture mechanics problems. Acta Mater 190:105–112

24. Yaghoobi M, Allison JE, Sundararaghavan V (2020) Multiscale modeling of twinning and detwinning behavior of HCP polycrystals. Int J Plastic 127:102653

25. Greeley D, Yaghoobi M, Pagan D, Sundararaghavan V, Allison J (2019) Using synchrotron radiation to improve understanding of deformation of polycrystalline metals by measuring, modelling and publishing 4D information. IOP Conf Ser: Mater Sci Eng 580:012017

26. Ganesan S, Yaghoobi M, Githens A, Chen Z, Daly S, Allison JE, Sundararaghavan V (2021) The effects of heat treatment on the response of WE43 Mg alloy: crystal plasticity finite element simulation and SEM-DIC experiment. Int J Plast 137:102917

27. Yaghoobi M, Chen Z, Sundararaghavan V, Daly S, Allison JE (2021) Crystal plasticity finite element modeling of extension twinning in WE43 Mg alloys: calibration and validation. Integr Mater Manuf Innov 10(3):488–507. https://doi.org/10.1007/s40192-021-00229-0

28. Yaghoobi M, Chen Z, Murphy-Leonard AD, Sundararaghavan V, Daly S, Allison JE (2022) Deformation twinning and detwinning in extruded Mg-4Al: in-situ experiment and crystal plasticity simulation. Int J Plast 155:103345

29. Yaghoobi M, Stopka KS, Lakshmanan A, Sundararaghavan V, Allison JE, McDowell DL (2021) PRISMS-Fatigue computational framework for fatigue analysis in polycrystalline metals and alloys. NPJ Computat Mater 7:38. https://doi.org/10.1038/s41524-021-00506-8

30. Stopka KS, Yaghoobi M, Allison JE, McDowell DL (2021) Effects of boundary conditions on microstructure-sensitive fatigue crystal plasticity analysis. Integr Mater Manuf Innov 10(3):393–412

31. Stopka KS, Yaghoobi M, Allison JE, McDowell DL (2022) Simulated effects of sample size and grain neighborhood on the modeling of extreme value fatigue response. Acta Mater 224:117524

32. Lakshmanan A, Yaghoobi M, Stopka KS, Sundararaghavan V (2022) Crystal plasticity finite element modeling of grain size and morphology effects on yield strength and extreme value fatigue response. J Market Res Technol 19:3337–3354

33. Bachmann F, Hielscher R, Schaeben H (2010) Texture analysis with MTEX–free and open source software toolbox. Trans Tech Publ 160:63–68

34. Yaghoobi M, Voyiadjis GZ, Sundararaghavan V (2021) Crystal plasticity simulation of magnesium and its alloys: a review of recent advances. Crystals 11(4):435

35. Tomé CN, Lebensohn RA, Kocks UF (1991) A model for texture development dominated by deformation twinning: application to zirconium alloys. Acta Metall Mater 39(11):2667–2680

36. Groeber MA, Jackson MA (2014) DREAM. 3D: a digital representation environment for the analysis of microstructure in 3D. Integr Mater Manuf Innov 3(1):56–72

37. Anand L, Kothari M (1996) A computational procedure for rate-independent crystal plasticity. J Mech Phys Solids 44(4):525–558

38. Voyiadjis G, Yaghoobi M (2019) Size effects in plasticity: from macro to nano. Academic Press, Cambridge

39. Bonet J, Wood RD (1997) Nonlinear continuum mechanics for finite element analysis. Cambridge University Press, Cambridge

40. Bronkhorst CA, Kalidindi SR, Anand L (1992) Polycrystalline plasticity and the evolution of crystallographic texture in FCC metals, philosophical transactions of the royal society of London. Series A: Phys Eng Sci 341(1662):443–477

41. Murphy-Leonard AD, Pagan DC, Beaudoin A, Miller MP, Allison JE (2019) Quantification of cyclic twinning-detwinning behavior during low-cycle fatigue of pure magnesium using high energy X-ray diffraction. Int J Fatigue 125:314–323

42. Qiao H, Agnew SR, Wu PD (2015) Modeling twinning and detwinning behavior of Mg alloy ZK60A during monotonic and cyclic loading. Int J Plast 65:61–84

43. Wu L (2009) Mechanical behavior and the role of deformation twinning in wrought magnesium alloys investigated using neutron and synchrotron x-ray diffraction. PhD dissertation, University of Tennessee, Knoxville

44. Wang H, Wu PD, Tomé CN, Wang J (2012) A constitutive model of twinning and detwinning for hexagonal close packed polycrystals. Mater Sci Eng, A 555:93–98

45. Arcari A (2010) Enhanced strain-based fatigue methodology for high strength aluminum alloys. PhD dissertation, Virginia Tech

46. Bozek JE, Hochhalter JD, Veilleux MG, Liu M, Heber G, Sintay SD, Rollett AD, Littlewood DJ, Maniatty AM, Weiland H, Christ RJ, Payne J, Welsh G, Harlow DG, Wawrzynek PA, Ingraffea AR (2008) A geometric approach to modeling microstructurally small fatigue crack formation: I. Probabilistic simulation of constituent particle cracking in AA 7075–T651. Modell Simulat Mater Sci Eng 16(6):065007

47. Kingma DP, Welling M (2013) Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114

48. Doersch C (2016) Tutorial on variational autoencoders, arXiv preprint arXiv:1606.05908

49. Chollet F (2016) Building autoencoders in keras, The Keras Blog 14

50. Zhang J, Joshi SP (2012) Phenomenological crystal plasticity modeling and detailed micromechanical investigations of pure magnesium. J Mech Phys Solids 60(5):945–972