Acta Materialia 284 (2025) 120537

Contents lists available at ScienceDirect

Acta Materialia

journal homepage: www.elsevier.com/locate/actamat

Active learning for the design of polycrystalline textures using conditional normalizing flows

Michael O. Buzzy [a], David Montes de Oca Zapiain [b], Adam P. Generale [c], Surya R. Kalidindi [a,c], Hojun Lim [b,*]

[a] School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[b] Computational Materials, Sandia National Laboratories, Albuquerque, NM 87185, USA
[c] George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

ARTICLE INFO

Keywords:
Active learning
Texture
Plastic anisotropy

ABSTRACT

Generative modeling has opened new avenues for solving previously intractable materials design problems. However, these new opportunities are accompanied by a drastic increase in the required amount of training data. This is in stark juxtaposition to the high expense and difficulty in curating such large materials datasets. In this work, we propose a novel framework for integrating generative models within an active learning loop. This enables the training of generative models with datasets significantly smaller than what has previously been demonstrated, providing a direct route for their application in data constrained environments. The functionality of this framework is then demonstrated by addressing the challenge of designing polycrystalline textures associated with target anisotropic mechanical properties. The developed protocol exhibited a cost reduction between 14 to 18 times over a randomly sampled experimental design.

1. Introduction

The development of computational algorithms/frameworks for performing materials design is a central goal of several recent and ongoing Materials Informatics and ICME (Integrated Computational Materials Engineering) efforts [1–9]. These algorithms facilitate the proposal of novel material internal structures corresponding to targeted properties of engineering interest (e.g., mechanical, thermal, magnetic, etc.) [10–15]. A subset of these algorithms are aimed at enabling microstructure sensitive design - where the goal is to specifically design a material's microstructure to meet the designer-specified combination of effective macroscale properties [1,10,12,16,17]. This task can be classified as an inverse problem, where the goal is to identify inverse solutions (predict microstructures given properties) given a predefined forward model (predict properties given microstructures). Due to extremely large and high-dimensional design spaces, the task of microstructure design constitutes a highly challenging inverse problem, even when there already exists an acceptable forward model [2,12]. In microstructure design, the forward models exhibit three characteristics that make inverting them challenging: (1) The forward mappings are typically ill-posed (i.e., many-to-one) meaning that multiple microstructures are expected to result in the same homogenized properties, resulting in non-unique solutions. (2) Various uncertainties prevail in the forward model, which reflect inadequate understanding of either the model forms or the values of the fitting parameters employed in the established forward model. (3) Descriptions of the microstructure are inherently high-dimensional, thereby presenting major challenges to the efficient exploration of the design space. Although notable advances have been made in this direction (e.g., property closures [18–20] and Bayesian optimization [21]), the approaches pursued in current literature do not adequately address the computational challenges involved in the efficient exploration of high-dimensional design spaces [18,22–24].

Bayesian inference [25–28] offers a practical avenue for overcoming the aforementioned challenges. Rather than attempting a deterministic inversion, Bayesian inference aims to identify a distribution of potential solutions, permitting a robust quantification of uncertainty of the variables involved. Most importantly, Bayesian methods allow for the incorporation of prior information into the inversion, providing critically needed regularization. Specifically, these approaches aim to solve inverse problems through the use of Bayes rule [26,28]:

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \tag{1}$$

where, $\mathbf{y}$ denotes an output of interest and $\mathbf{x}$ is the corresponding input of a noisy mapping (e.g., $y = f(x) + \epsilon$). Bayesian inference makes the inversion tractable, despite the inversion being an ill-posed problem,

* Corresponding author.
E-mail address: hnlim@sandia.gov (H. Lim).

https://doi.org/10.1016/j.actamat.2024.120537
Received 18 July 2024; Received in revised form 30 September 2024; Accepted 3 November 2024
Available online 12 November 2024
1359-6454/Published by Elsevier Ltd on behalf of Acta Materialia Inc.

by limiting the search space through the regularization via the specification of the prior $p(\mathbf{x})$. Traditional algorithms for performing Bayesian inference (such as Markov Chain Monte Carlo [29–34]) have struggled to address the challenges of high-dimensionality, while optimization-based variational inference approximations circumvent this limitation, they have traditionally been severely limited in their flexibility, due to the assumed form of the target posterior [35]. Only recently have novel generative modeling frameworks made Bayesian inference tractable in high-dimensional spaces [2,3,12]. In particular, recent advances have demonstrated the unique capabilities of generative models in the formal stochastic treatment of the mesoscale details of heterogeneous material systems [2,36–40]. These efforts have made clear that such generative models offer a viable strategy capable of addressing all three challenges outlined previously. Detrimentally, the incorporation of these models demands a significant increase in the amount of training data required for establishing the desired mappings in these high-dimensional spaces. Consequently, the use of generative models for designing microstructures has been limited to applications where a large number of simulations are readily available; recent works have used 11,000 [3] to 70,000 [1] numerical simulations. Such large dataset requirements impede the adoption of generative models to problems with elevated computational, or physical costs of dataset generation (e.g., complex physics-based simulations [41,42] or sophisticated experiments [43–45]). Our main goal in this work is to explore active learning strategies towards enabling a dramatic reduction in the data requirements for producing reliable generative models.

We formulate materials design as learning the generative inverse model, $f(\cdot)$, which approximates the distribution $p(\mathbf{m}|\mathbf{P})$, where we define $\mathbf{m}$ as a suitable (frequently high-dimensional) representation of the materials' internal structure, and $\mathbf{P}$ a set of properties of interest. It is highly desirable to explore opportunities to actively learn $f(\cdot)$ because of the high computational cost of producing the training data [23,46]. However, typical protocols involved in active learning break down in the inverse direction. More specifically, the steps involved include (i) querying the current version of the model being learned on a large number of candidate inputs, (ii) applying a selection criteria to determine the specific candidate input that exhibits the highest expected potential for improving the model fidelity, and (iii) interrogating the selected points to generate new ground truth data, which is then added to the training dataset [47]. The three-step active learning process described above is then repeated until a stopping criteria is met, indicating a saturation point past which the incorporation of additional data provides minimal novel information to the underlying model.

The execution of these steps becomes intractable when attempting to learn inverse mappings such as $f(\cdot)$, as a byproduct of the ill-posed nature of such inverse mappings in which multiple or no solutions may exist [48]. This poses direct hurdles in executing steps (i) and (iii) of active learning described above, and indirect hurdles in executing step (ii). For example, the query on a potential candidate input can produce distinct multiple outputs, raising questions regarding the application of a selection criteria [47]. Furthermore, knowledge regarding ground truth function evaluations can often only be interrogated in the forward direction. In other words, even if a desirable candidate point for training is identified, we often have no way of evaluating the ground truth in the inverse direction to establish a new reliable training data point. As a direct consequence of these challenges, novel strategies are needed to objectively select new candidates for training $f(\cdot)$.

In an effort to circumvent the hurdles described above, we turn to a novel paradigm pursued in recent works [2,12]. Here, a surrogate model (typically a Gaussian process), $g(\cdot)$, substitutes the ground truth forward process, i.e., $g(\cdot) \approx p(\mathbf{P}|\mathbf{m})$. The inverse generative model, $f(\cdot)$, is then trained on a synthetic dataset produced by repeatedly sampling the surrogate throughout its domain [2,49]. These studies have also demonstrated that replacing $g(\cdot)$ for the ground truth does not significantly impact the accuracy of $f(\cdot)$ [2]. Essentially, this approach seeks to establish approximations to both conditional distributions of interest, $p(\mathbf{P}|\mathbf{m})$ and $p(\mathbf{m}|\mathbf{P})$. Previous work by Generale et al. [2] chose to learn the functions $g(\cdot)$ and $f(\cdot)$ sequentially. In other words, they first trained $g(\cdot)$ in isolation, then subsequently trained $f(\cdot)$. This sequential approach results in the function $g(\cdot)$'s estimation of $p(\mathbf{P}|\mathbf{m})$ being static throughout the training of $f(\cdot)$. In contrast, we aim to simultaneously learn both forward and inverse maps, providing dynamic refinement of $p(\mathbf{P}|\mathbf{m})$ and $p(\mathbf{m}|\mathbf{P})$, respectively, for any iterative cycle of the training process. These dynamic estimates are then utilized to implement novel and suitably-designed active learning strategies for efficient learning of both the forward and inverse maps. Specifically, we propose to implement a *selection criteria* defined on $p(\mathbf{P}|\mathbf{m})$ (based on typical metrics such as maximum variance), combined with a *stopping criteria* defined on $p(\mathbf{m}|\mathbf{P})$ (based on stabilization of the posterior density).

The novel framework proposed and demonstrated in this work effectively combines data-efficient approaches for learning forward models with the expressive power of generative modeling for inverse solutions. Specifically, the proposed framework exhibits the following desired features: (i) **Data Efficiency** - the framework employs a novel active learning strategy for optimal training of desired mappings with sparse datasets in regimes with high costs of data curation; (ii) **High Dimensionality** - inference remains feasible across high-dimensional microstructure spaces; (iii) **Probabilistic** - the model can quantify uncertainty and is flexible enough to capture potentially complex multimodal distributions of inverse solutions.

The example application selected for the present study using the proposed framework involves the design of polycrystalline microstructures exhibiting a targeted set of anisotropic plasticity properties. More specifically, features of the polycrystalline microstructures considered in the design process are restricted to the crystallographic orientations (i.e., texture) in the sample, while the plastic properties are taken to be directional yield stress and lateral strain increment values that could be used to define the anisotropic yield surface of the material. The forward relationships between the crystallographic texture and their anisotropic mechanical response have been studied extensively in prior literature [13,50–55]. However, performing design for this class of problems has remained challenging due to the high-dimensionality, ill-posedness, and non-linearity involved. For the selected case study, the material microstructure (i.e., texture) is represented using the first 38 coefficients of the Generalized Spherical Harmonic (GSH) expansion [56], and the plastic response of interest is represented using 54 anisotropy parameters (18 stresses and 36 strain increments) [52,57]. For the desired high-fidelity forward and inverse maps between these high-dimensional spaces, the use of the proposed framework resulted in an improvement of a factor between 14 to 18 in data efficiency over random sampling . Thus enabling the establishment of an inverse model in a data-efficient and data-driven manner.

2. Theoretical background

2.1. Texture and plastic anisotropy

The distribution of crystal lattice orientations in polycrystalline metallic samples (i.e., texture) is one of the main contributors to their anisotropic plastic response [15,19,52,57–61]. This is largely a consequence of the fact that the dominant mechanism of plastic deformation in most metals is dislocation slip, occurring exclusively on certain preferred planes and directions in a single crystal [50]. Plastic anisotropy is critical in accurate predictions of formability and structural failure in sheet metals, and is typically described by continuum-scale phenomenological yield models [59,62,63]. For establishing the surrogate models of interest for the present study, we first must adopt suitable representations for both texture and plastic anisotropy.

Mathematically, texture is quantified by 1-point statistics of the crystal lattice orientations in the sample, using a function known as the orientation distribution function (ODF). The crystal lattice orientation, ζ, at any point in a polycrystalline sample can be represented by a

set of three Bunge–Euler angles, i.e., $\zeta = (\varphi_1, \Phi, \varphi_2)$ [10,56]. The ODF expresses the probability density over the orientation space in a given polycrystalline sample. Any function over the orientation space can be represented using a basis known as Generalized Spherical Harmonics (GSH) [10,56]. Specifically, the ODF, denoted as $\mathcal{F}(\zeta)$, can be expanded as

$$\mathcal{F}(\zeta) = \sum_{l=0}^{\infty} \sum_{u=1}^{U(l)} \sum_{v=1}^{N(l)} C_l^{uv} \dot{T}_l^{uv}(\zeta) \tag{2}$$

where, C_l^{uv} represents the spectral coefficients and $\dot{T}_l^{uv}$ denote suitably symmetrized GSH basis reflecting the symmetries arising from both the single crystal and sample scale considerations [56].

An important property of the GSH expansion is that single crystal microstructures are represented as impulse distributions. The GSH coefficients for the single crystal microstructures are then expressed as [56]

$$C_l^{uv} = (2l+1)\dot{T}_l^{*uv}(\zeta_o) \tag{3}$$

where, ζ_o denotes the lattice orientation in such microstructures, and * denotes the complex conjugate. Since the ODF of any polycrystal microstructure can be expressed as a weighted sum of single crystal ODFs, one can show that the GSH space corresponding to all physically realizable ODFs is in fact convex [10]. This property of the GSH representations will become critical later when defining the selection candidate pool in this work. The GSH expansion shown in Eq. (2) contains an infinite number of terms, and thus in practice must be truncated. Following prior work on polycrystalline microstructures exhibiting cubic-orthorhombic symmetries, we consider only the first 38 terms of this expansion in this work [20,52,57], which corresponds to a truncation level of $l = 12$.

The classical approach for modeling macroscale metal plasticity involves utilization of a yield criterion that describes the onset of plastic flow, the evolution of the yield surface with plastic deformation, and the direction of the plastic flow (flow rule). Using this approach, a considerable number of plastic anisotropic yield models have been developed [64–68]. These constitutive models are parameterized by directionally-dependent parameters such as yield strengths and/or r-values. However, these macroscale yield models do not directly incorporate the effect of crystallographic texture on the resultant anisotropic response of the metal alloy. In this work, we are primarily interested in capturing the influence of texture on the anisotropic yield surface for the polycrystalline microstructure, as predicted by advanced computational micromechanical models such as the crystal plasticity (CP) models [57,69,70].

Following the success of previous work [57], we characterize the influence that texture has on the anisotropic behavior of a polycrystalline materials by using a set of normalized yield stresses and normalized lateral plastic strain increments. Specifically, these quantities were obtained from CP simulations [55,57], defining the ground truth forward model as $g^*(\cdot)$ throughout this work. Function evaluations of this model correspond to imposing a plastic work increment of 0.4 MPa under uniaxial tension conditions in a selected set of sample directions. The tensile loading directions were selected at 15 degrees increments in the $X-Y$, $Y-Z$ and $Z-X$ planes (a total of 18 directions). At each tensile direction k, a yield stress is extracted and normalized by the yield stress in the X direction. The normalized yield stress for a direction k is denoted by $\sigma_{norm}^{o,k}$. Additionally, at each direction k, two lateral plastic strain increments were also obtained. The two lateral plastic strain increments were similarly normalized so that the product plastic work increment becomes the unit value of 1. The normalized lateral plastic strain increments for a tensile direction k are denoted by $\delta_{inc_{1,2}}^{k}$. Therefore, each microstructure has a total of 54 anisotropy parameters, which correspond to 18 normalized yield stresses and 36 normalized lateral plastic strain increments. A detailed description of the crystal plasticity framework and simulation procedure can be found in [57]. This dataset consisted of 13,408 single crystal and 61,482 polycrystalline textures, and their associated values of $\sigma_{norm}^{o,k}$ and $\delta_{inc_{1,2}}^{k}$. This large collection of ground truth data is invaluable in evaluating the efficacy of our active learning framework. While our active learning strategy will ultimately use only a fraction of this dataset, this large dataset allows us to train a second generative model across the entirety of the available dataset, serving as a point of reference for our framework moving forward. Establishing this baseline is key as it permits a more rigorous evaluation of solution posterior distributions throughout the active learning process.

2.2. Active learning

With the inputs and outputs defined, we now turn our attention to actively learning models for the forward and inverse relationships. All active learning efforts require three key components: (i) the ability to generate a candidate pool of potential experiments to run, (ii) a suitable form for the surrogate model, and (iii) a selection criteria compatible with the surrogate model. Our choices for each of these requirements are described in the following sections.

2.2.1. Candidate pool generation

Approaches for generating candidates broadly divides active learning into three distinct classes: pool-based, stream-based, and membership query synthesis [47]. Given our existing dataset, this study leverages pool-based active learning schemes, where a set of candidates is defined *a priori* (in contrast to the other two methods where candidates are created throughout the learning process). In pool-based active learning schemes, the candidate set denotes all possible inputs (microstructures) which could be interrogated to obtain ground truth outputs (properties). In the pursuit of the best design solutions, it would be ideal for the candidate pool to fill the entire space of possible inputs. This is key for two reasons. First, by covering the input space, we implicitly cover the output space, allowing our active learning strategy to potentially explore the entire space of achievable properties. Second, the generative model's capacity to identify a diverse set of inverse solutions is inherently tied to the diversity of the candidate pool. From a Bayesian perspective, the candidate pool represents a prior $p(\mathbf{m})$ for the generative model, enforcing background knowledge regarding the inversion [2,12]. Revisiting Bayes rule, $p(\mathbf{m}|\mathbf{P}) \propto p(\mathbf{P}|\mathbf{m})p(\mathbf{m})$, we can see that our generative model (i.e., approximation to $p(\mathbf{m}|\mathbf{P})$) will be unlikely to propose new inverse solutions outside the prior where $p(\mathbf{m})$ is small. A tight prior could therefore limit the generative model from proposing otherwise acceptable inverse solutions.

Given this background, we turn our primary focus towards obtaining a diverse candidate pool covering the complete input space. Looking more closely at our GSH representations of texture, we note that the space of valid GSH coefficients forms a convex set constructed via the interpolation of single crystal impulse ODFs [10,56]. This property of the GSH is critical for two reasons: (i) It allows us to approximately define the boundary of the candidate space, thus elucidating what candidates are acceptable (i.e., $p(\mathbf{m})$ is zero for points outside the boundary). (ii) It allows us to sample from the candidate space (i.e., through interpolations of points on the boundary). These properties of the GSH allow for the creation of a diverse candidate pool which nearly fills the entirety of the input space. Prior works have similarly followed this procedure to create space filling designs over the GSH convex hull. In this work, we will utilize the dataset from Park et al. [52,57] given the fact that they thoroughly sample the GSH convex hull. It is important to note that the creation of space-filling candidate pools is a non-trivial task. This is particularly true for higher-order microstructure statistics which go beyond the ODF, which remains a current area of active research [8,36–39].

2.2.2. Multi-output Gaussian process regression

The next step is to identify a surrogate forward modeling strategy that is well suited for active learning. Within pool-based active learning schemes, there are multiple choices of algorithms based on probabilistic approaches [71,72], committee methods [73,74], and others [47]. The corresponding options for surrogate modeling are also varied (e.g., Gaussian Process Regression [75], Bayesian neural networks [76,77], Committee neural networks [73] and Random Forest Committees [26]). Taking into account the constraints of the problem at hand such as data efficiency, uncertainty quantification, and the high non-linearity of the forward model, the most suitable choice for the selected application is a probabilistic active learning scheme utilizing Gaussian Process Regression (GPR). In particular, Gaussian Processes (GPs) have been shown to be effective learners on small datasets [75], have varied established active learning criteria [47], and offer uncertainty quantification [75].

The forward model $g(\cdot)$ aims to approximate the likelihood, $g(\cdot) \approx p(\mathbf{P}|\mathbf{m})$. Leveraging GPs, we represent $g(\cdot) = \mathcal{GP}(v(\cdot), k(\cdot,\cdot'))$, where $v(\cdot)$ is the mean function, and $k(\cdot,\cdot')$ is the covariance function. Typically, a zero-mean approximation, i.e., $v(\cdot) = 0$, is invoked, consistent with the typical practice of normalizing all input and output variables to exhibit zero means and unit variances (i.e., z-scores [75]). The covariance function aims to identify the similarity or correlation between any two points in microstructure space $\mathbf{m}$ and $\mathbf{m}'$. The common Radial Basis Function (RBF) with Automatic Relevance Determination (ARD) is used in this work [75]:

$$k(\mathbf{m},\mathbf{m}') = exp\left(-\frac{1}{2}\sum_{j=1}^{D}\frac{1}{l_j^2}(m_j - m_j')^2\right) \tag{4}$$

where, D is the dimensionality of the microstructure representation $\mathbf{m}$, and l_j is a hyper-parameter representing the characteristic length-scale of each dimension. Since our application requires consideration of possibly correlated outputs (i.e., $\mathbf{P} \in \mathbb{R}^H$), we leverage Multi-Output Gaussian Processes (MOGPs) based on the Linear Method of Coregionalization (LMC), which constructs the multi-output function via a linear transformation $\mathbf{W} \in \mathbb{R}^{H\times L}$ of L latent Gaussian processes, $q(\mathbf{m}) = \{\mathcal{GP}(0, k_l(\cdot,\cdot'))\}_{l=1}^{L}$, each with its own covariance function. The forward model is then expressed as:

$$g(\mathbf{m}) = \mathbf{W}q(\mathbf{m}) \tag{5}$$

2.2.3. Selection criteria

As the last step of the active learning algorithm, we define a selection criteria to identify the most promising candidates for the sequential generation of new training data points. Specifically, this requires the adoption and computation of a scalar metric for each member of the candidate pool. A large variety of selection criteria have been employed in prior literature [47]. For the present application, we must employ a selection criteria that is compatible with Gaussian Process Regression — our choice for building the forward model. The simplest approach for the present application would be to use the Maximum Variance (MV) criterion, as this information is already computed in the GPR. Specifically, the MV criteria selects the next training point as the candidate that exhibits the highest prediction uncertainty estimated by the GPR. The MV criterion has been observed to work well when the noise in the forward model is roughly uniform throughout the input domain [47]. For the present application, since the MOGPs prediction variance will be a matrix, we adopt its determinant as the selection criterion.

2.3. Generative modeling: Conditional normalizing flows

Next, we turn our attention towards generative modeling strategies which will identify the desired inverse solutions. Prior efforts in materials design have explored variational autoencoders (VAEs) [12], diffusion models [1], normalizing flows [3], continuous normalizing flows [2], and generative adversarial networks (GANs) [78]. For this work, we opted to use conditional normalizing flows (cNFs) due to their rigorous probabilistic formulation (necessary for accurate density estimation) [79], efficient training and evaluation times (critical when the model will be trained repeatedly in an active learning loop) [80], and easy likelihood computations [35,81]. The goal of our generative model is to approximate the distribution $p(\mathbf{m}|\mathbf{P})$. A cNF aims to approximate distributions through the change of variables formula [80]

$$p(\mathbf{m}|\mathbf{P},\theta) = p_z((f_\theta^{\mathbf{P}})^{-1}(\mathbf{m}))\left|\det\left(\frac{\partial f_\theta^{\mathbf{P}^{-1}}}{\partial \mathbf{m}}\right)\right| \tag{6}$$

where, the cNF is an invertible neural network[1] $f_\theta^{\mathbf{P}} : \mathbf{z} \rightarrow \mathbf{m}$, where latent variable $\mathbf{z}$ is assumed to exhibit a tractable probability distribution (e.g., unit-Gaussian) enabling exact likelihood evaluation. In this notation, it is implied that the inverse $(f_\theta^{\mathbf{P}})^{-1} : \mathbf{m} \rightarrow \mathbf{z}$ is known and parameterized by the same tunable parameters θ used to describe the forward mapping. Note that both the forward and inverse maps are conditioned on the specified value of the properties of interest $\mathbf{P}$, which are allowed to vary within the prior [80]. Through this mapping, we gain the ability to reason with the complex conditional distribution $p(\mathbf{m}|\mathbf{P})$ in two ways. (1) We can evaluate the likelihood of a point in the microstructure space for a specified $\mathbf{P}$ by passing values of $\mathbf{m}$ and $\mathbf{P}$ through Eq. (6) [35,81]. (2) We can generate samples from $p(\mathbf{m}|\mathbf{P})$ by sampling $\mathbf{z}$ from the unit Gaussian and passing it through $f_\theta^{\mathbf{P}}$ along with the specified $\mathbf{P}$. This is the property that allows the cNF to perform generation (i.e., the ability to propose novel designs) [80].

The use of an invertible neural network architecture is paramount as the mapping can be learned in one direction, and its inverse immediately obtained for free. Furthermore, we are able to train the model via a maximum likelihood loss expressed as [80]

$$\mathcal{L}_{ml} = \mathbb{E}\left[\frac{1}{2}\|(f_\theta^{\mathbf{P}})^{-1}(\mathbf{m})\|_2^2 - \log|\mathbf{J}|\right] \tag{7}$$

where, $\mathbf{J} = \det(\frac{\partial (f_\theta^{\mathbf{P}})^{-1}}{\partial \mathbf{m}})$. The use of this loss function results in a regularized objective which resists over-fitting [35]. It should be noted that the computation of $\log|\mathbf{J}|$ is non-trivial and potentially computationally expensive. There exists a large body of work on formulating invertible architectures that allow for easy computation of this term [35,81,82]. Our choices for model architecture are outlined in Appendix A.

In data scarce applications, training via Eq. (7) poses a significant challenge, as only a limited number of paired values of $\mathbf{m}$ and $\mathbf{P}$ might be available. In the present application, the situation is even more dire, as the candidate pool only contains microstructural information without any property information. With the goal of utilizing cNFs on smaller datasets, recent works have addressed this challenge by first training a surrogate forward model on a small dataset and then using that surrogate to generate a larger synthetic dataset, suitable for training the cNF [2,83,84]. These works have shown empirically that the dataset size required for calibrating the forward model is less than that required to calibrate an inverse model. Furthermore, they suggest that this substitution leads to a negligible performance loss in the final cNF — an observation we will evaluate critically in our work.

3. Proposed framework

In this section, we formally present our proposed framework. Similar to all active learning frameworks, the goal is to minimize data requirements by only simulating points of high informational value.

[1] Prior efforts have expressed cNFs as $\mathbf{m} = f_\theta(\mathbf{z},\mathbf{P})$, and the corresponding inverse as $\mathbf{z} = f_\theta^{-1}(\mathbf{m},\mathbf{P})$. We find this notation misleading. Since both these functions take in $\mathbf{P}$, they are not technically inverses of each other. Furthermore, in practice $\mathbf{P}$ behaves more like a parameter of the network than an input. Therefore, we have chosen to write the conditioning variable $\mathbf{P}$ as a superscript rather than an input.

However, unlike traditional active learning efforts, our final goal is not to just learn a surrogate forward model, but also to simultaneously construct a model which learns a sufficiently accurate probabilistic inverse mapping suitable for materials design efforts. As already mentioned, this poses a significant challenge because the execution of traditional active learning algorithms becomes intractable when attempting to learn inverse mappings due to their inherently ill-posed nature [48]. In an effort to overcome this challenge, we propose a modified active learning framework that employs suitable metrics on both the forward and inverse mappings. Specifically, we propose to combine a selection criteria on the forward model along with a stopping criteria on the inverse model. The proposed framework broadly follows the five step process depicted in Fig. 1 in each iteration of the active learning loop: (i) Forward model calibration, (ii) Synthetic dataset creation, (iii) Inverse model calibration, (iv) Stopping criterion check, and (v) Data acquisition using a selection criterion.

In **Step 1**, we calibrate the forward MOGP as described in Section 2.2.2. To perform this calibration, we use a dataset of ground-truth observations (for the present problem, this is obtained from simulations performed using crystal plasticity models). This dataset will be denoted as $\mathcal{G}^* = \{\mathbf{m}_o, g^*(\mathbf{m}_o)\}_{o=0}^{G}$, where g^* represents the true forward model, and the microstructures $\mathbf{m}_o$ are a subset of the candidate pool (the subset for which the crystal plasticity model has been evaluated) $\mathcal{M} = \{\mathbf{m}_i\}_{i=0}^{M}$ established before the start of the active learning loops. Any new ground-truth data acquired in the active learning loops (**Step 5**) will be continuously added to the dataset $\mathcal{G}^*$. The main goal of the active learning algorithm is to keep the size of $\mathcal{G}^*$ small as the ground-truth data generation is presumed to be computationally expensive. In order to ensure there is data to calibrate the MOGP in the first active learning iteration, we initialize $\mathcal{G}^*$ with a small number of ground-truth observations. We found our algorithm to be effective with as few as 5 ground-truth observations for the initial set.

As mentioned in the previous section, training a cNF via Eq. (7) requires a large dataset. Clearly, it would be infeasible to train the cNF using the limited available ground-truth data (recall that our goal is to keep this dataset as small as possible). In recent work [2], this hurdle was mitigated by exploiting the ability of MOGPs to produce a large synthetic dataset at low computational cost which approximates the ground-truth forward data. The impact of utilizing this approximation will be investigated in later sections. We expect that the accuracy of such a synthetically generated training dataset for the cNF model would improve as more ground-truth data is added. In **Step 2**, we produce the synthetic data set $\mathcal{S}$ needed to train the cNF model (i.e., the inverse map), while accounting for the uncertainty in the MOGP predictions. In our proposed framework, we accomplish this task by sampling multiple data points from the posterior prediction of properties for each microstructure in the candidate pool $\mathcal{C}$. The synthetically produced dataset $\mathcal{S}$ can then be expressed as

$$\mathcal{S} = \{\mathbf{m}_{s\%M}, \mathbf{P}_s\}_{s=0}^{(M*S)-1} \tag{8}$$

where, $\mathbf{P}_s$ is a sample from $g(\mathbf{m}_{s\%M}) \approx p(\mathbf{P}|\mathbf{m}_{s\%M})$, S is the number of samples drawn from the MOGP prediction of property corresponding to each microstructure, and % is the modulo operator. Since the computational cost of building this synthetic dataset is quite low, we used a relatively high value of $S = 100$ in this work. In addition to constructing the synthetic dataset in **Step 2**, we also store the associated uncertainties produced by the GP in the covariance matrices $\mathbf{\Sigma}_i$ for every $\mathbf{m}_i$ from the candidate pool $\mathcal{M}$, since this will be required later in **Step 5**.

Next, in **Step 3**, we calibrate the cNF $f_\theta^{\mathbf{P}}$ using the synthetic dataset $\mathcal{S}$ using the process described in Section 2.3. **Step 4** checks if the active learning process should continue using a stopping criteria, which will be defined in the next section. Finally, if the stopping criteria is not met, **Step 5** acquires a new data point from the physics-based crystal plasticity model selected via the maximum variance criteria described in Section 2.2.3. The new observation is then added to $\mathcal{G}^*$. Once the active learning process has terminated, the results will include both a calibrated MOGP (i.e., forward map) and a calibrated cNF $f_\theta^{\mathbf{P}}$ (i.e., inverse map), which can be used in downstream design tasks.

It is emphasized here that the proposed active learning framework is designed to be highly data efficient by simultaneously training both the forward and inverse maps in each active learning iteration. A second distinctive feature of our proposed framework lies in the use of the MOGP to produce the dataset needed to train the cNF. This is critical because the synthetic dataset $\mathcal{S}$ can be made arbitrarily large by increasing the number of structures in the candidate pool, $\mathcal{M}$, or increasing the number of samples taken from the posterior of the MOGP.

3.1. Identifying a stopping criterion

Of the four requirements for active learning (candidate pool, ground truth model, selection criteria, and stopping criteria), we have yet to identify a suitable stopping criteria. The goal of a stopping criteria is to identify when the active learning process is no longer yielding new or valuable information. Since our goal is materials design, the stopping criteria should be defined on the inverse model. In other words, it should indicate when our ability to perform materials design has converged. Here we propose a novel stopping criteria suitable for the active learning of cNFs.

The key property of cNFs we will exploit to define a stopping criteria is their ability to perform direct likelihood computations. Given a cNF that may or may not be well calibrated, we can pass a validation dataset of ground-truth observations $\mathcal{V} = \{\mathbf{m}_a, \mathbf{P}_a\}_{a=0}^{V}$ through Eq. (6). In establishing the validation set $\mathcal{V}$, we have selected microstructures distinct from those in the candidate pool $\mathcal{C}$. In this work, we created a validation dataset $\mathcal{V}$ using $V = 50$ microstructures. Assuming independence between the data points, we can compute the likelihood of the validation set, given the model parameters θ, as

$$p(\mathcal{V}|\theta) = \prod_{n=0}^{A} p(\mathbf{m}_a|\mathbf{P}_a, \theta) \tag{9}$$

As we update the parameters θ within the active learning loops, we expect the likelihood of the validation dataset to increase, indicating that we are more closely approximating the true distribution $p(\mathbf{m}|\mathbf{P})$.[2] Therefore, we define our stopping criteria to track the value of the (log-)likelihood to signal when it ceases to improve with continued active learning iterations. In our work, we have employed the following stopping criterion:

$$\frac{1}{K}\sum_{k=I-K+1}^{I} |p(\mathcal{V}|\theta_k) - p(\mathcal{V}|\theta_{k-1})| < \phi \tag{10}$$

where θ_k denotes the parameters associated with active learning iteration k, I denotes the total number of completed active learning iterations, and K is a hyperparameter denoting the number of steps we wish to average over. Specifically, we employed $K = 5$, and $\phi = 0.001$ in this work.

4. Results

4.1. Active learning performance

In order to evaluate critically the efficacy of our proposed active learning approach, we built and compared the performances of three different cNFs. The first cNF, referred as cNF Full, is trained on a dataset in which the entire pool of 54,480 textures were simulated with the ground truth crystal plasticity model. The second, referred as cNF Random, is trained on randomly selected subsets of this same dataset.

[2] As is typical for numerical stability we work with the log of this likelihood in practice.

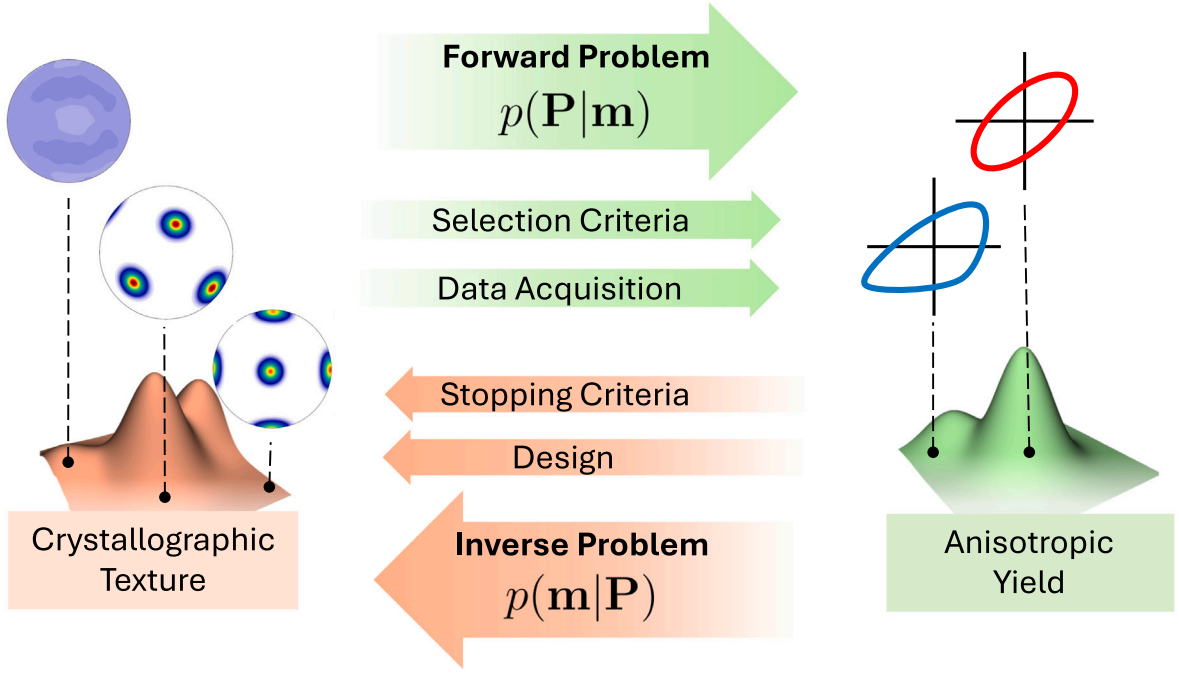

Fig. 1. A schematic of the proposed active learning framework. Here we update both the forward and inverse maps within each active learning iteration.

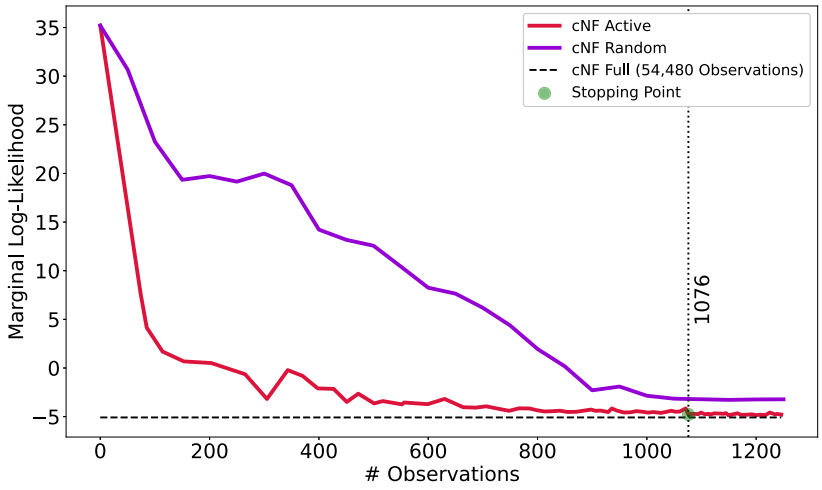

Fig. 2. Comparison of the performance of the three cNFs produced in this work — cNF Full, cNF Random, and cNF Active. The likelihood values for the test set are plotted. The likelihood values for the three cNF models at the stopping point are −3.5 for cNF Random, −4.8 for cNF Active, and −5.0 for cNF Full.

The third, referred as cNF Active, is trained using the active learning approach developed in this work. cNF Full establishes the lower bound for the fidelity of the other cNFs learned using the different learning frameworks. In other words, as the number of observations increases, the fidelity of the different learning frameworks should converge to the performance of cNF Full. Our objective is therefore to approach this lower bound as quickly as possible, i.e., with the fewest number of observations. Conversely, cNF Random denotes a threshold trajectory past which an active learning framework provides no additional value over purely random selection. As such, the efficacy of an active learning framework can be readily assessed by examining how it outperforms random selection. In order to ensure a fair comparison, the cNF Random results presented in this work are shown after averaging over 50 different initializations. Furthermore, we compute and compare the likelihood of the different models on the test set (20,000 ground truth observations), which indicates how probable the learned approximation of $p(\mathbf{m}|\mathbf{P})$ by each model is for the given ground truth test set. Therefore, the likelihood provides a comparison of the accuracy of the trained inverse models after each active learning iteration, and thereby allows us to critically evaluate the efficacy of the active learning algorithm.

As shown in Fig. 2, active learning provides significant advantage in both low data (<900 observations) and high data (> 900 observations) regimes. In the low data regime, we see that active learning dramatically outperforms random sampling, as expected. With so few observations, each observation has a significant impact on the overall performance of the model. Active learning therefore offers tremendous benefits in this regime. Eventually, after a sufficient number of observations have been obtained, the random sampling performance begins to catch up to the active learning strategy. In the present example, this occurs at around 900 observations with the stopping point occurring at 1076 observations. Although the performance difference between the active and random strategies appears relatively small in this high data region, it is notable that the loss function used, the marginal log-likelihood, is a logarithmic measure of accuracy. Therefore the performance difference between the Random and Active models is still significant despite their visual convergence. Importantly, we note that

at the stopping point cNF Random has failed to achieve a suitable level of accuracy to be useful for downstream design tasks. A more detailed comparison of the three models accuracy at the stopping point can be seen in Appendix B. Furthermore, we observe that the random selection approach does not reach comparable performance relative to the active learning strategy at the stopping point until approximately 19,000 observations (This convergence can be see in Appendix C). In the active learning approach, the stopping criterion was attained after 1,076 observations (c.f. to Fig. 2). Therefore, given our choice of stopping criteria, the proposed active learning approach needed approximately 14 to 18 times fewer observations than random sampling to achieve roughly the same model accuracy. However, it is clear from Fig. 2, that the realized level of data savings is a function of the stopping criterion. In other words, if we were to stop much earlier (and accept a significant drop in accuracy in favor of a lower data budget) we would observe that the difference between the Random and Active approaches would be less pronounced. Minimally, for any stopping criteria, we see that the active learning approach outperforms random sampling by 2 times,[3] and conversely we see a 20 times data savings at their maximal difference. The "stalling" behavior of the random approach (seen beyond 900 observations) is likely because the remaining observations are not statistically diverse from the large number of preexisting data points the model has already observed. Therefore, intelligently identifying the remaining data points which still offer novel information is what allows the active strategy to much more rapidly converge to peak performance.

Importantly, we observe that the active learning approach does indeed converge to the performance of the cNF trained on the full dataset. We also see that the convergence, for this problem, is rapid and generally smooth. Combined these observations empirically suggest (i) the design pattern of training a cNF using a synthetic dataset generated by an MOGP is effective, and results in little loss in accuracy when compared to using ground truth data directly (Additionally shown in Appendix B), and (ii) the active learning process is stable, in that we can expect the addition of new data to result in a more accurate model. These findings are critical in enabling the practical use of cNF's across a wider array of problems in data scarce applications.

4.2. Model accuracy

Now that we have shown that our active learning strategy can effectively learn a cNF on a limited dataset, we turn our attention to evaluating its accuracy. The nMLL in the previous section serves as an ideal measure of accuracy as it takes into account the stochastic nature of the cNF. However, it fails to provide any intuitive sense of how well the model is performing. Therefore, we investigate model accuracy via two other metrics. First, we will use a "round trip" error which will enable us to quantify performance using familiar visualizations and metrics. Second, to better understand the learned posteriors, we will take select properties from the test set ($\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$) and plot their posterior distributions $p(\mathbf{m}|\mathbf{P}_i)$ along with the ground truth values of $\mathbf{m}$.

The round trip accuracy starts with a target property value $\mathbf{P}^*$, then utilizes the Active cNF $f(\cdot)$ to predict a posterior distribution of microstructures $p(\mathbf{m}^*|\mathbf{P}^*)$ given the design target. This distribution is then sampled, and the samples are evaluated through the forward model $g(\cdot)$ to obtain $p(\mathbf{P}|\mathbf{m}^*)$. We then compare the expectation of this distribution to the target property value $\mathbf{P}^*$. The round trip accuracy is needed because bijectivity is not guaranteed in inverse problems. There may be multiple solutions $\mathbf{m}^*$ which lead to a single $\mathbf{P}^*$. In this way, the error between the known $\mathbf{m}$ from the test set and the solution from the cNF $\mathbf{m}^*$ may be vastly different even though they yield the same $\mathbf{P}^*$. The round trip error circumvents this issue by checking if the proposed solutions in the microstructure space have the desired properties which were initially targeted. Thus, this round trip incorporates the degeneracy of the forward model into the error metric. The round trip error also has the added benefit of benchmarking the entire framework, both forward and inverse model, at the same time. Furthermore, the round trip error can be visualized for the entire test set at once, giving us a holistic sense of the model performance over a large number of samples. We compute the round trip error for all 20,000 data points in the test set. Fig. 3 shows the aggregated results for the round trip errors for the textures present in the train and the test set. The overall errors are within respectable margins. The parity plots in Fig. 3 show $\mathbf{P}^*$ versus the round trip prediction, and have a strong linear trend. It is important to point out that the parity plot shows all the values for the 18 normalized yield stresses (denoted by σ^o_{norm}) for each of the textures of present in the train and test set accordingly in a single plot to make it easy to visualize the performance across all the normalized stresses. This is done similarly for the lateral plastic strain increments where all the 36 lateral plastic strain increments (denoted by δ_{inc}) for each texture of present in the train and test set are shown in a single plot. Lastly, the CDF plot shows that 95 percent of predicted data exhibits a % Error less than 4.2 percent.

In an effort to evaluate the prediction accuracy of specific points, results from three selected target properties, $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$, are taken from the test set. These test cases were chosen because, in their totality, they form a diverse set of points with respect to the many of the measures used in this work, and therefore should serve well to evaluate critically the general performance of the model. The test cases differ in three notable ways. Firstly, they differ in the input space, with texture ranging from randomly textured ($\mathbf{P}_1$) to highly textured ($\mathbf{P}_2$ and $\mathbf{P}_3$) (see Fig. 4). Consequentially, they also span the output property space ranging from nearly isotropic ($\mathbf{P}_1$) to highly anisotropic ($\mathbf{P}_2$ and $\mathbf{P}_3$). Finally, they span the space in terms of % Error ranging from accurate points in the test set ($\mathbf{P}_1$, $\mathbf{P}_2$, < 1% error) to the least accurate ($\mathbf{P}_3$ ~ 10% error). cNF Active was used to determine the posteriors $p(\mathbf{m}|\mathbf{P}_i)$ for each property.

The results from these case studies are shown in Fig. 4. It is clear that the model has produced reasonable results compared to known values of $\mathbf{m}$ for the chosen values of $\mathbf{P}$. For the isotropic design target, $\mathbf{P}_1$, we see a nearly random texture was predicted (shown in the pole figure in Fig. 4), and for the highly anisotropic samples we see corresponding highly textured predictions. Looking more closely at the posteriors, we see good agreement with the ground truth values from the test set. All of the ground truth values are within the predicted posteriors, and are typically near high density regions. In addition, posteriors from least textured to most (top to bottom) show a clear trend in the width of the posteriors. For the nearly randomly textured point, $\mathbf{P}_1$, the posteriors are very tight (± 0.02) while highly textured samples exhibit wider distributions. For $\mathbf{P}_3$, the posteriors have a width of ± 0.2. Comparing the two highly textured data points $\mathbf{P}_2$ and $\mathbf{P}_3$, we see that despite similar shaped posteriors, the posterior estimates for $\mathbf{P}_2$ have their peaks align better with the ground truth values than $\mathbf{P}_3$. Overall, these posteriors corroborate the excellent model performance shown via the other metrics, and further elucidate how the model performance varies depending on which textures are being predicted.

5. Conclusions

Active learning has the attractive property of significantly reducing model training data requirements, enabling the production of optimal experimental designs. Coupling such active training procedures with generative modeling serves as an promising avenue for solving inverse design problems in costly and data-scarce regimes. In this work, we have proposed a framework which actively learns a cNF to solve inverse material design problems. We find that our approach is able to reduce the amount of data required to train cNFs by 14 to 18 times, while maintaining a high level of accuracy. In addition to data efficiency,

[3] Here we ignore the starting point where both models are trained on the same initial observations, and therefore have the same accuracy.

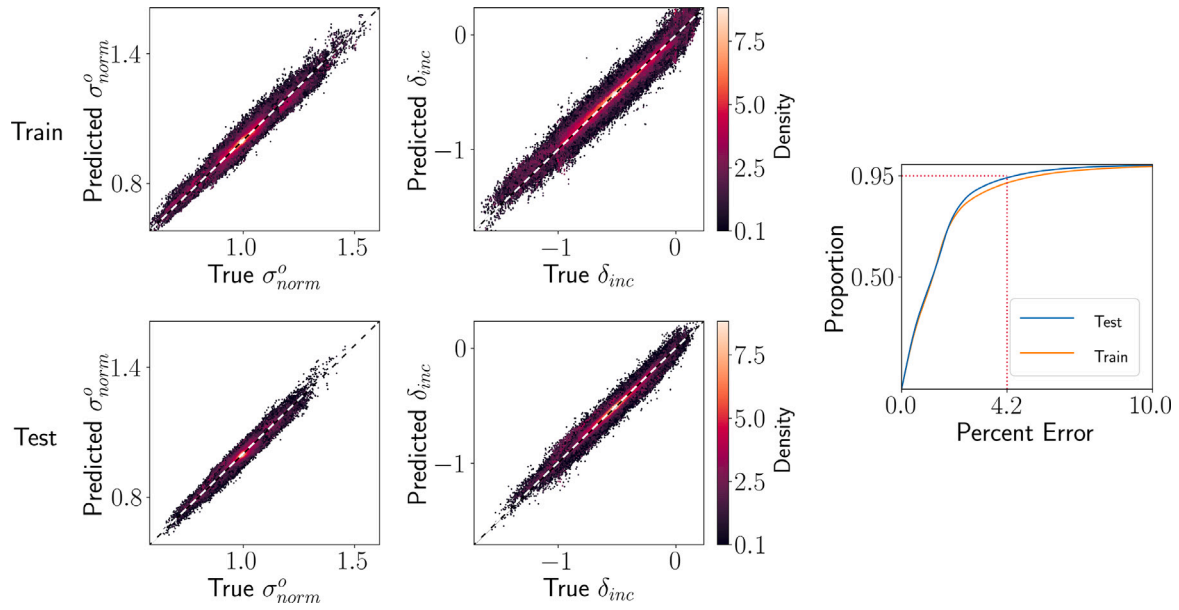

Fig. 3. The round trip performance is evaluated. Parity plots are shown for both the train and test set. The 54 anisotropy parameters are grouped into normalized yield stresses (σ^o_{norm}) and strain increments (δ_{inc}) and are shown across two parity plots. Due to the high number of points, they are colored based on their density to better show arrangement. The cumulative distribution function (CDF) plot on the right shows that for the test set, 95% of the data exhibits less than 4.2% Error$_{RT}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

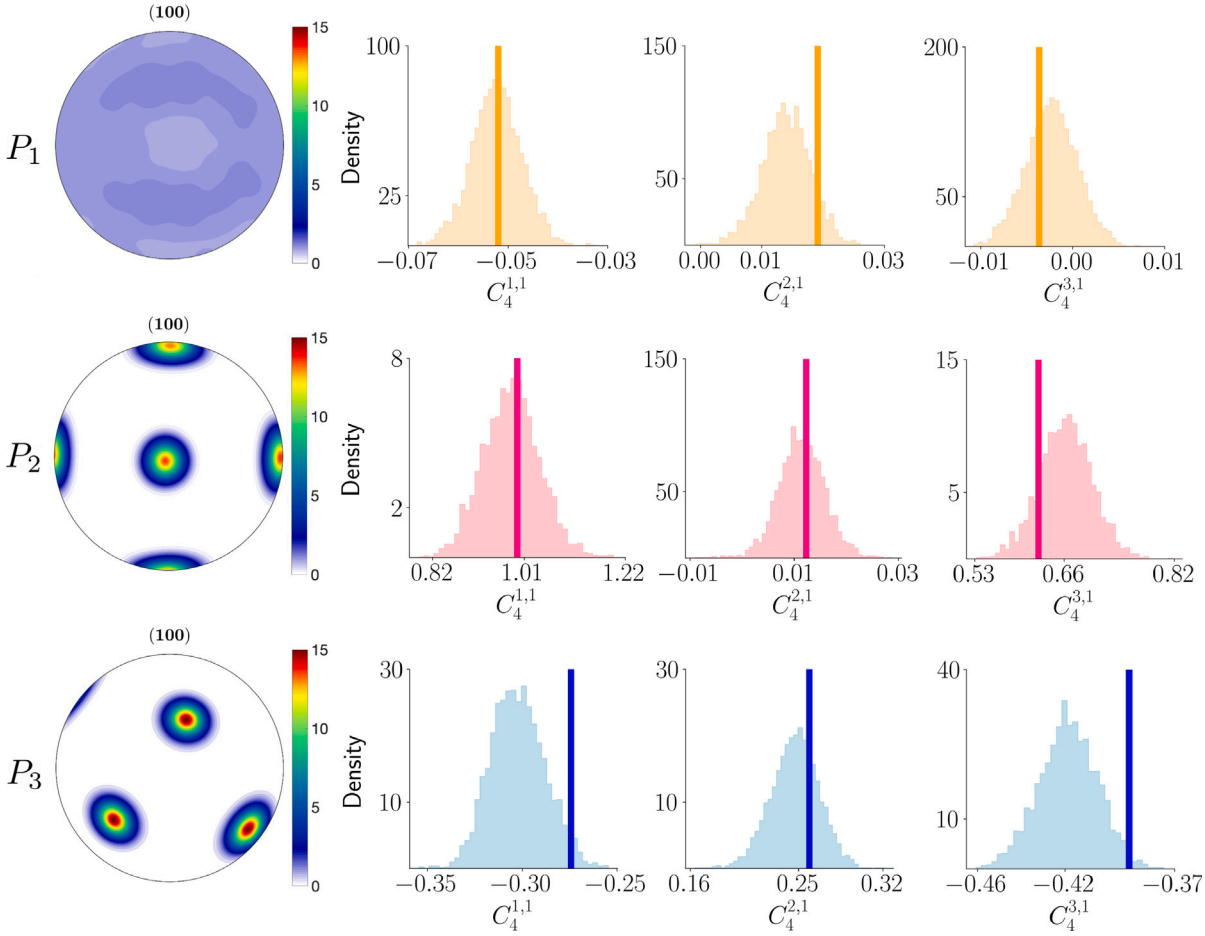

Fig. 4. The posterior distributions of **m** are plotted for selected properties $\mathbf{P}_1$, $\mathbf{P}_2$, and $\mathbf{P}_3$. For better visualization, we exploit the natural ordering of the GSH representation of **m** and plot only the first three GSH coefficients ($C_4^{1,1}, C_4^{2,1}, C_4^{3,1}$) as they form the first and most dominate truncation level of the GSH expansion used in this work. The textures are also visualized as pole figures on the left where higher values signify more dominate orientations. Each row shows the pole figures (corresponding to the MAP values of the posterior) and the posterior distributions for each point. The bold vertical lines are the ground truth values of **m**.

we also demonstrated that our framework returns accurate inverse solutions, as well as being able to quantify the various uncertainties involved.

It is important to note that while our framework is data efficient, it is only data efficient in the requirement of labeled input/output data pairs. Depending on the size of the cNF, it may still be necessary to have a large number of unlabeled data points in the input space. For many scientific and engineering problems, such as the one presented here, obtaining unlabeled data points is trivial. In our case, the GSH space is well defined, and it is easy to generate a large numbers of samples. However, this is not generally true. For example, problems that involve difficult to obtain data such as micrographs [12], may not be able to produce enough samples to train the cNF. For these problems, new methods are emerging in microstructure generation which may provide a viable avenue to execute this type of framework [36–38,85], and will be the focus of future work. Currently, the concepts in this paper are generally applicable only when large amounts of unlabeled inputs are available, but obtaining their corresponding outputs is expensive.

CRediT authorship contribution statement

Michael O. Buzzy: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **David Montes de Oca Zapiain:** Writing – review & editing, Writing – original draft, Supervision. **Adam P. Generale:** Formal analysis. **Surya R. Kalidindi:** Writing – review & editing, Supervision. **Hojun Lim:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge Natalie Mata, Andreas Robertson, Ian Slagle, Rafael Orozco, and Aditya Venkatraman for their support and feedback in the production of this manuscript.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration, USA under contract DE-NA0003525.

Appendix A. Model architectures/training

A.1. Gaussian process regression

For the forward model we used a multi-output GPR with correlated outputs using the linear method of coregionilaztion. We implemented this GPR with the package *GPyTorch* [86] as it utilizes PyTorch as its back-end and enables convenient gradient calculations of the forward model. The kernel used in the GPR was the common radial basis function (RBF) which was additively combined with a white noise kernel. Each input dimension possessed independent length scale terms, which were obtained by maximum likelihood estimation [26]. The average training time for one active learning iteration was 1-2 min.

A.2. Normalizing flow

Our inverse model is a cNF which is composed of RealNVP coupling blocks [81]. We created a network consisting of 20 invertible blocks. Within each block we utilized a 2 layer fully connected neural network with 128 neurons in each layer with sigmoid activation's. Our implementation is based on the *FrEIA* package [87] which provides ready to use invertible neural network modules in *PyTorch*. We trained for 70 epochs using the Adam optimizer with a learning rate of .01, and a batch size of 200. The training pool consisted of 54,480 samples. The experiments were performed on a Linux server running RHEL 8 with 16 Xeon cores and 4x Nvidia P100 GPUs. The average training time for one active learning iteration was 10–12 min.

Appendix B. Comparisons of model performance

Here we show the relative performance of cNF Active, Random, and Full at the stopping point of 1076 observations. Fig. B.5 shows the error metrics for each model. Clearly we can see that despite the similar likelihood values of each model at the stopping point (−4.8 for cNF Active, −3.5 for cNF Random, and −5.0 for cNF Full), there still exists a substantial difference between the predictive performance of each model — particularly for cNF Random. With respect to their cumulative distribution functions we can see that the predictions of cNF Random are far less accurate than cNF Active. The percent error which encompasses 95% of the data is 32% for cNF Random relative to 4.2% for cNF Active. Clearly, this large increase in error would likely make the predictions from cNF Random unusable for any downstream tasks. Furthermore, we additionally show the posterior of $C_4^{3,1}$ for the point of interest P_2 in Fig. B.5. Here we immediately see a shift in the mean of the posterior away from the known ground truth value. This corroborates the reduction in performance seen in the cumulative distribution function. Additionally, we also observe a difference in the widths of the predicted posteriors. The posterior of cNF full is the tightest with its mean centered at the known ground truth value form the test set. The posterior of cNF Active is located in alignment with the posterior of cNF Full, but is slightly wider. This is due to the additional uncertainty introduced via the approximation of the forward process via a Gaussian Process. Overall, the increase in uncertainty is negligible. In addition to having its mean no longer located near the known ground truth value, the posterior of cNF Random is very wide — nearly encompassing half of the entire acceptable range of the value of $C_4^{3,1}$. This indicates that cNF Random is highly uncertain in its prediction, and this widened posterior is consistently observed across many data points in the test set. Clearly, this large difference in performance further substantiates the claim that the Active learning framework has calibrated a performant model with significantly less data than would be required with a random selection process.

Appendix C. cNF random — high data regime convergence

Here Fig. C.6 shows the convergence of the cNF Random to the performance of cNF Active for the high data regime. To better illustrate the high data regime convergence we have elected to plot the mean round trip percent error, $\overline{\%Error_{RT}}$, for each model rather than the marginal log likelihood used in Fig. 2. This is because the $\%Error_{RT}$ is no longer a logarithmic measure of accuracy (where the marginal log likelihood is), and therefore produces a more informative visualization in the high data region. We would like to note that the convergence point is roughly the same for both measures of performance. Inspecting Fig. C.6 we can see that with respect to $\overline{\%Error_{RT}}$ cNF Random gradually approaches the performance of cNF Active (3.348), and that the two curves meet around 19,000 observations. Generally, we see that the performance of the two models begins to be comparable around 14,000 observations despite not matching exactly. The slow convergence exhibited here showcases the claim that the active learning framework is 14 to 18 times more data efficient than the random selection process.

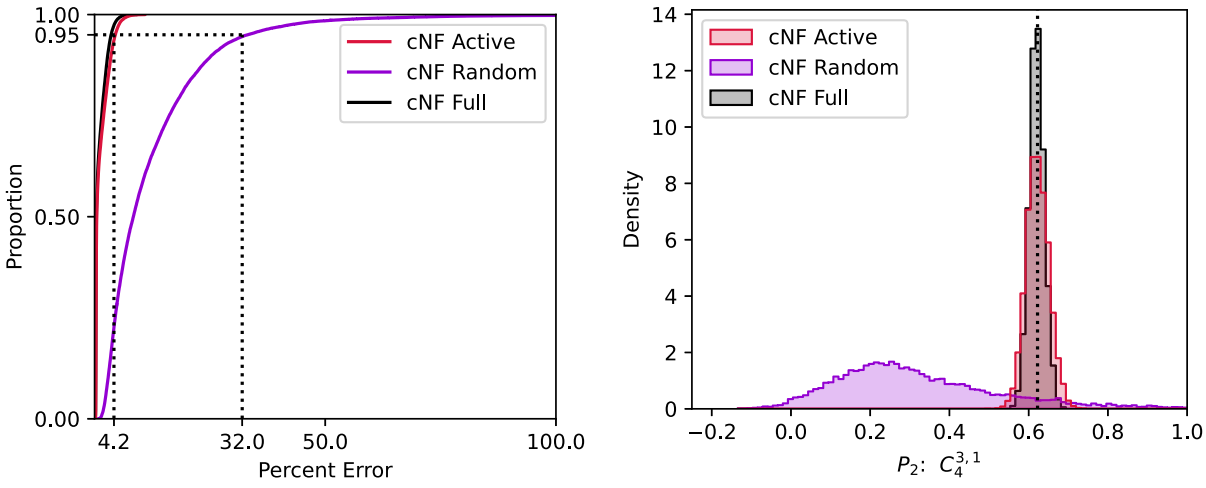

Fig. B.5. Comparison of the accuracy of the three cNFs (cNF Active, cNF Random, and cNF Full) at the stopping point of 1076 observations. The left plot shows the CDF of the error for each model. Here we can see the percent error required to encompass 95 Percent of the data shifts from 4.2%, for cNF Active, to 32% for cNF Random. The right plot shows the predicted posterior densities for the value of $C_4^{3,1}$ for the point of interest P_2. The ground truth value from the test set is shown via the dotted vertical line.

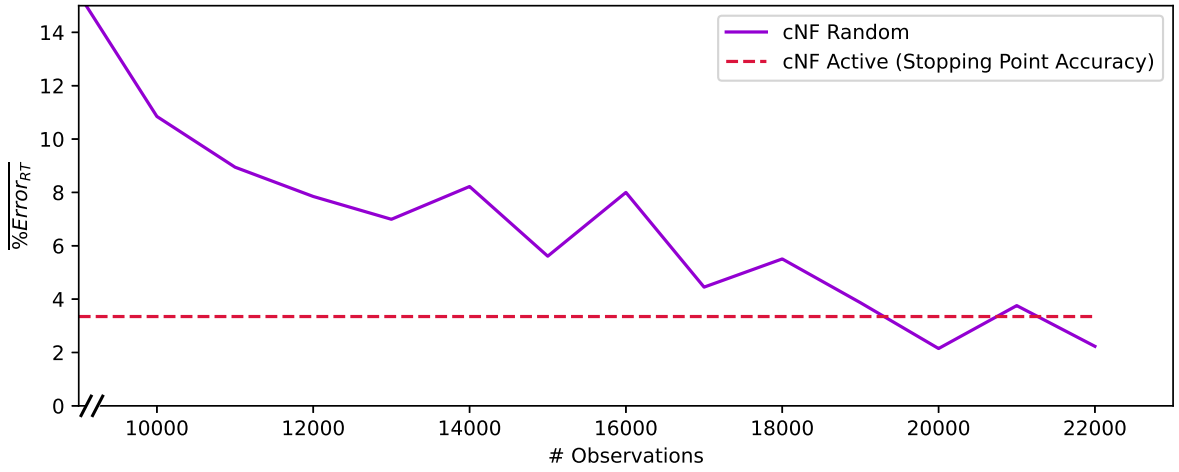

Fig. C.6. The convergence of cNF Random to the performance of cNF Active at the stopping criteria. Please note performance on this plot with respect to the average $\%Error_{RT}$, not the log-likelihood as in Fig. 2. We observe a roughly linear improvement in performance with increasing observations, and see cNF Random converge to cNF active (stopping point) around 19,000 observations. The performance of cNF random was evaluated at 1000 observation increments past 10,000 observations to limit computational cost.

References

[1] N.N. Vlassis, W. Sun, Denoising diffusion algorithm for inverse design of microstructures with fine-tuned nonlinear material properties, Comput. Methods Appl. Mech. Engrg. 413 (2023) 116126, http://dx.doi.org/10.1016/j.cma.2023.116126.

[2] A. Generale, C. Kelly, G. Harrington, A. Robertson, M. Buzzy, S. Kalidindi, A Bayesian approach to designing microstructures and processing pathways for tailored material properties, in: AI for Accelerated Materials Design - NeurIPS 2023 Workshop, 2023, URL https://openreview.net/forum?id=zZPICTs5gB.

[3] V. Fung, J. Zhang, G. Hu, P. Ganesh, B.G. Sumpter, Inverse design of two-dimensional materials with invertible neural networks, npj Comput. Mater. 7, 200, http://dx.doi.org/10.1038/s41524-021-00670-x.

[4] A. Merchant, S. Batzner, S.S. Schoenholz, M. Aykol, G. Cheon, E.D. Cubuk, Scaling deep learning for materials discovery, Nature (2023) 1–6.

[5] N. Brodnik, C. Muir, N. Tulshibagwale, J. Rossin, M. Echlin, C. Hamel, S. Kramer, T. Pollock, J. Kiser, C. Smith, S. Daly, Perspective: Machine learning in experimental solid mechanics, J. Mech. Phys. Solids 173 (2023) 105231, http://dx.doi.org/10.1016/j.jmps.2023.105231.

[6] D.M. Dimiduk, E.A. Holm, S.R. Niezgoda, Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering, Integr. Mater. Manuf. Innov. 7 (2018) 157–172.

[7] M. Diehl, M. Groeber, C. Haase, D.A. Molodov, F. Roters, D. Raabe, Identifying structure–property relationships through DREAM. 3D representative volume elements and DAMASK crystal plasticity simulations: an integrated computational materials engineering approach, Jom 69 (2017) 848–855.

[8] S.R. Kalidindi, M. Buzzy, B.L. Boyce, R. Dingreville, Digital twins for materials, Front. Mater. 9 (2022) 818535.

[9] D.L. McDowell, S.R. Kalidindi, The materials innovation ecosystem: a key enabler for the materials genome initiative, Mrs Bull. 41 (4) (2016) 326–337.

[10] B. Adams, S. Kalidindi, D. Fullwood, Microstructure Sensitive Design for Performance Optimization, Butterworth-Heinemann, Waltham, MA, 2013.

[11] P. Acar, V. Sundararaghavan, Stochastic design optimization of microstructural features using linear programming for robust design, AIAA J. 57 (2019) 448–455.

[12] A. Generale, A. Robertson, C. Kelly, S. Kalidindi, Inverse stochastic microstructure design, 2023, http://dx.doi.org/10.2139/ssrn.4590691, SSRN: Preprint.

[13] C. Alleman, J. Foulk, A. Mota, H. Lim, D. Littlewood, Concurrent multiscale modeling of microstructural effects on localization behavior in finite deformation solid mechanics, Comput. Mech. 61 (1–2) (2018) 207–218.

[14] T. Fast, S.R. Kalidindi, Formulation and calibration of higher-order elastic localization relationships using the MKS approach, Acta Mater. 59 (2011) 4595–4605, http://dx.doi.org/10.1016/j.actamat.2011.04.005.

[15] X. Wu, G. Proust, M. Knezevic, S. Kalidindi, Elastic–plastic property closures for hexagonal close-packed polycrystalline metals using first-order bounding theories, Acta Mater. 55 (8) (2007) 2729–2737.

[16] Y. Gao, Y. Liu, Relibaility-based topology optimization with stochastic heterogeneous microstructure properties, Mater. Des. (2021) http://dx.doi.org/10.1016/j.matdes.2021.109713.

[17] J. Jung, J.I. Yoon, H.K. Park, H. Jo, H.S. Kim, Microstructure design using machine learning generated low dimensional and continuous design space, Materialia 11 (2020) 100690.

[18] A. Mann, S. Kalidindi, Development of a robust CNN model for capturing microstructure-property linkages and building property closures supporting material design, Front. Mater. 9 (2022) 851085, http://dx.doi.org/10.3389/fmats.2022.851085.

[19] M. Knezevic, S.R. Kalidindi, R.K. Mishra, Delineation of first-order closures for plastic properties requiring explicit consideration of strain hardening and crystallographic texture evolution, Int. J. Plast. 24 (2) (2008) 327–342.

[20] G. Proust, S.R. Kalidindi, Procedures for construction of anisotropic elastic–plastic property closures for face-centered cubic polycrystals using first-order bounding relations, J. Mech. Phys. Solids 54 (8) (2006) 1744–1762.

[21] D. Khatamsaz, B. Vela, P. Singh, D. Johnson, D. Allaire, R. Arroyave, Bayesian optimization with active learning of design constraints using an entropy based approach, NPJ: Comput. Mater. 9 (2023) 1–14, http://dx.doi.org/10.1038/s41524-023-01006-7.

[22] C. Chen, G. Gu, Generative deep neural networks for inverse materials design using backpropagation and active learning, Adv. Sci. 7 (2020) http://dx.doi.org/10.1002/advs.201902607.

[23] D. Khatamsaz, B. Vela, P. Singh, D.D. Johnson, D. Allaire, R. Arróyave, Multi-objective materials bayesian optimization with active learning of design constraints: Design of ductile refractory multi-principal-element alloys, Acta Mater. 236 (2022) 118133, http://dx.doi.org/10.1016/j.actamat.2022.118133.

[24] J. Dornheim, L. Morand, S. Zeitvogel, T. Iraki, N. Link, D. Helm, Deep reinforcement learning methods for structure-guided processing path optimization, J. Intell. Manuf. 33 (2022) http://dx.doi.org/10.1007/s10845-021-01805-z.

[25] A. Castillo, A. Venkatraman, S. Kalidindi, Mechanical responses of primary-α Ti grains in polycrystalline samples: Part II – Bayesian estimation of the crystal-level elastic-plastic mechanical properties from spherical-indentation measurements, Integr. Mater. Manuf. Innov. 10 (2021) 99–114, http://dx.doi.org/10.1007/s40192-021-00204-9.

[26] C.M. Bishop, N.M. Nasrabadi, Pattern Recognition and Machine Learning, vol. 4, (no. 4) Springer, 2006.

[27] R.D. Wilkinson, Bayesian Calibration of Expensive Multivariate Computer Experiments, in: L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, Y. Marzouk, L. Tenorio, B. Van Bloemen Waanders, K. Willcox (Eds.), Large-Scale Inverse Problems and Quantification of Uncertainty, John Wiley & Sons, Ltd, Chichester, UK, 2010, pp. 195–215, http://dx.doi.org/10.1002/9780470685853.ch10, URL https://onlinelibrary.wiley.com/doi/10.1002/9780470685853.ch10.

[28] G. James, D. Witten, T. Hastie, R. Tibshirani, et al., An Introduction to Statistical Learning, vol. 112, Springer, 2013.

[29] M. Betancourt, A conceptual introduction to Hamiltonian Monte Carlo, 2017, arXiv preprint arXiv:1701.02434.

[30] C. Andrieu, J. Thoms, A tutorial on adaptive MCMC, Stat. Comput. 18 (2008) 343–373.

[31] A. Shahmoradi, F. Bagheri, J.A.e. Osborne, Fast fully-reproducible serial/parallel Monte Carlo and MCMC simulations and visualizations via ParaMonte Python library, 2020, arXiv e-prints arXiv:2010.00724.

[32] J.S. Speagle, A conceptual introduction to Markov chain Monte Carlo methods, 2019, arXiv preprint arXiv:1909.12313.

[33] M.K. Cowles, B.P. Carlin, Markov chain Monte Carlo convergence diagnostics: a comparative review, J. Am. Stat. Assoc. 91 (434) (1996) 883–904.

[34] R. Bardenet, A. Doucet, C. Holmes, Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach, in: International Conference on Machine Learning, PMLR, 2014, pp. 405–413.

[35] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: International Conference on Machine Learning, PMLR, 2015, pp. 1530–1538.

[36] M.O. Buzzy, A.E. Robertson, S.R. Kalidindi, Statistically conditioned polycrystal generation using denoising diffusion models, Acta Mater. (2024) 119746.

[37] A.E. Robertson, C. Kelly, M. Buzzy, S.R. Kalidindi, Local–global decompositions for conditional microstructure generation, Acta Mater. 253 (2023) 118966, http://dx.doi.org/10.1016/j.actamat.2023.118966.

[38] A. Robertson, S. Kalidindi, Efficient generation of N-field microstructures from 2-point statistics using multi-output Gaussian random fields, Acta Mater. 232 (2022) 117927, http://dx.doi.org/10.1016/j.actamat.2022.117927.

[39] A.E. Robertson, A.P. Generale, C. Kelly, M.O. Buzzy, S.R. Kalidindi, MICRO2D: A large, statistically diverse, heterogeneous microstructure dataset, Integr. Mater. Manuf. Innov. (2024) 1–35.

[40] N.N. Vlassis, W. Sun, Denoising diffusion algorithm for inverse design of microstructures with fine-tuned nonlinear material properties, Comput. Methods Appl. Mech. Engrg. 413 (2023) 116126, http://dx.doi.org/10.1016/j.cma.2023.116126.

[41] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. In't Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, et al., LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, Comput. Phys. Comm. 271 (2022) 108171.

[42] S.L. Kramer, A. Jones, A. Mostafa, B. Ravaji, T. Tancogne-Dejean, C.C. Roth, M.G. Bandpay, K. Pack, J.T. Foster, M. Behzadinasab, et al., The third sandia fracture challenge: predictions of ductile fracture in additively manufactured metal, Int. J. Fract. 218 (2019) 5–61.

[43] M.D. Uchic, M.A. Groeber, A.D. Rollett, Automated serial sectioning methods for rapid collection of 3-D microstructure data, Jom 63 (2011) 25–29.

[44] N.R. Brodnik, C.-J. Hsueh, K.T. Faber, B. Bourdin, G. Ravichandran, K. Bhattacharya, Guiding and trapping cracks with compliant inclusions for enhancing toughness of brittle composite materials, J. Appl. Mech. 87 (3) (2020) 031018.

[45] H. Ding, V. Fridrici, G. Guillonneau, S. Sao-Joao, J. Geringer, J. Fontaine, P. Kapsa, Investigation on mechanical properties of tribofilm formed on Ti–6Al–4V surface sliding against a DLC coating by nano-indentation and micro-pillar compression techniques, Wear 432 (2019) 202954.

[46] N. Wilson, D. Willhelm, X. Qian, R. Arróyave, X. Qian, Batch active learning for accelerating the development of interatomic potentials, Comput. Mater. Sci. 208 (2022) 111330, http://dx.doi.org/10.1016/j.commatsci.2022.111330.

[47] B. Settles, Active Learning Literature Survey, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[48] P. Putzky, M. Welling, Recurrent inference machines for solving inverse problems, 2017, http://dx.doi.org/10.48550/arXiv.1706.04008, ArXiV.

[49] S.T. Radev, U.K. Mertens, A. Voss, L. Ardizzone, U. Köthe, BayesFlow: Learning complex stochastic models with invertible neural networks, 2020, arXiv:2003.06281.

[50] S.R. Kalidindi, C.A. Bronkhorst, L. Anand, Crystallographic texture evolution in bulk deformation processing of FCC metals, J. Mech. Phys. Solids 40 (3) (1992) 537–569.

[51] S.R. Kalidindi, Incorporation of deformation twinning in crystal plasticity models, J. Mech. Phys. Solids 46 (2) (1998) 267–290.

[52] D. Montes de Oca Zapiain, H. Lim, T. Park, F. Pourboghrat, Predicting plastic anisotropy using crystal plasticity and Bayesian neural network surrogate models, Mater. Sci. Eng. A 833 (2022) 142472, http://dx.doi.org/10.1016/j.msea.2021.142472, URL https://www.sciencedirect.com/science/article/pii/S0921509321017342.

[53] H. Lim, J.D. Carroll, C.C. Battaile, T.E. Buchheit, B.L. Boyce, C.R. Weinberger, Grain-scale experimental validation of crystal plasticity finite element simulations of tantalum oligocrystals, Int. J. Plast. 60 (2014) 1–18.

[54] H. Lim, C.C. Battaile, J.D. Carroll, B.L. Boyce, C.R. Weinberger, A physically based model of temperature and strain rate dependent yield in BCC metals: Implementation into crystal plasticity, J. Mech. Phys. Sol. 74 (2015) 80–96.

[55] T. Park, H. Lim, B. Reedlunn, S. Kramer, E. Corona, F. Pourboghrat, The impact of heterogeneous microstructural features on crystal plasticity modeling of plastic anisotropy, Modelling Simul. Mater. Sci. Eng. Submitted (2021).

[56] H.J. Bunge, Texture Analysis in Materials Science: Mathematic Methods, Butterworth & Co, Berlin, 1982.

[57] T. Park, D. Montes de Oca Zapiain, F. Pourboghrat, H. Lim, Development of a deep learning model for capturing plastic anisotropy–texture linkage, JOM 75 (12) (2023) 5466–5478.

[58] N. Tardif, S. Kyriakides, Determination of anisotropy and material hardening for aluminum sheet metal, Int. J. Solids Struct. 49 (25) (2012) 3496–3506.

[59] E. Corona, S.L.B. Kramer, W.M. Scherzinger, A.R. Jones, Anisotropic plasticity model forms for extruded Al 7079: Part I, calibration, Int. J. Solids Struct. 213 (2021) 135–147.

[60] S. Kalidindi, S. Schoenfeld, On the prediction of yield surfaces by the crystal plasticity models for fcc polycrystals, Mater. Sci. Eng. A 293 (1–2) (2000) 120–129.

[61] T. Fast, M. Knezevic, S.R. Kalidindi, Application of microstructure sensitive design to structural components produced from hexagonal polycrystalline metals, Comput. Mater. Sci. 43 (2) (2008) 374–383.

[62] M. Baral, T. Hama, E. Knudsen, Y.P. Korkolis, Plastic deformation of commercially-pure titanium: experiments and modeling, Int. J. Plast. 105 (2018) 164–194.

[63] F. Barlat, J. Brem, J.W. Yoon, K. Chung, R. Dick, D. Lege, F. Pourboghrat, S.-H. Choi, E. Chu, Plane stress yield function for aluminum alloy sheets—part 1: theory, Int. J. Plast. 19 (9) (2003) 1297–1319.

[64] A.S. Khan, S. Huang, Continuum Theory of Plasticity, John Wiley & Sons, 1995.

[65] R. Hill, A theory of the yielding and plastic flow of anisotropic metals, Proc. R. Soc. London. Ser. A. Math. Phys. Sci. 193 (1033) (1948) 281–297.

[66] O. Cazacu, B. Plunkett, F. Barlat, Orthotropic yield criterion for hexagonal closed packed metals, Int. J. Plast. 22 (7) (2006) 1171–1194.

[67] H. Choi, J.W. Yoon, Stress integration-based on finite difference method and its application for anisotropic plasticity and distortional hardening under associated and non-associated flow rules, Comput. Methods Appl. Mech. Engrg. 345 (2019) 123–160.

[68] M.H. Ulz, C.C. Celigoj, A uniquely defined multiplicative elasto-plasticity model with orthotropic yield function and plastic spin, Comput. Methods Appl. Mech. Engrg. 374 (2021) 113565.

[69] K. Zhang, B. Holmedal, O. Hopperstad, S. Dumoulin, Modelling the plastic anisotropy of aluminum alloy 3103 sheets by polycrystal plasticity, Modelling Simul. Mater. Sci. Eng. 22 (7) (2014) 075015.

[70] C. Liu, M. Li, T. Yue, Thick anisotropy analysis for AA7B04 aluminum plate using CPFEM and its application for springback prediction in multi-point bending, Int. J. Adv. Manuf. Technol. 115 (4) (2021) 1139–1153.

[71] D.M. de Oca Zapiain, A. Venkatraman, R. Katona, D. Maestas, M. Roop, P. Noell, An active learning framework for the rapid assessment of galvanic corrosion, 2024.

[72] D.A. Cohn, Z. Ghahramani, M.I. Jordan, Active learning with statistical models, J. Artif. Intell. Res. 4 (1996) 129–145.
[73] L. Morand, N. Link, T. Iraki, J. Dornheim, D. Helm, Efficient exploration of microstructure-property spaces via active learning, Front. Mater. 8 (2022) 824441.
[74] S. Kee, E. Del Castillo, G. Runger, Query-by-committee improvement with diversity and density in batch active learning, Inform. Sci. 454 (2018) 401–418.
[75] C. Williams, C. Rasmussen, Gaussian processes for regression, in: Advances in Neural Information Processing Systems, vol. 8, 1995.
[76] V. Mullachery, A. Khera, A. Husain, Bayesian neural networks, 2018, arXiv preprint arXiv:1801.07710.
[77] A. Venkatraman, D. Montes de Oca Zapiain, S.R. Kalidindi, Reduced-order models for ranking damage initiation in dual-phase composites using Bayesian neural networks, JOM 72 (2020) 4359–4369.
[78] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, J. Hu, Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials, npj Comput. Mater. 6 (1) (2020) 84.
[79] I. Kobyzev, S.J. Prince, M.A. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 3964–3979.
[80] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, U. Köthe, Guided image generation with conditional invertible neural networks, 2019, CoRR abs/1907.02392, arXiv:1907.02392.
[81] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, 2017, arXiv:1605.08803.
[82] D.P. Kingma, P. Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, in: Advances in Neural Information Processing Systems, vol. 31, 2018.
[83] S. Radev, U. Mertens, A. Voss, L. Ardizzone, U. Kothe, BayesFlow: Learning complex stochastic models with invertible neural networks, IEEE Trans. Neural Netw. Learn. Syst. 33 (2022) 1452–1466, http://dx.doi.org/10.1109/TNNLS.2020.3042395.
[84] R. Orozco, F.J. Herrmann, P. Chen, Probabilistic Bayesian optimal experimental design using conditional normalizing flows, 2024, arXiv:2402.18337.
[85] P. Seibert, M. Husert, M. Wollner, K. Kalina, M. Kastner, Fast reconstruction of microstructures with ellipsoidal inclusions using analytic descriptors, 2023, http://dx.doi.org/10.48550/arXiv.2306.08316, ArXiv.
[86] J.R. Gardner, G. Pleiss, D. Bindel, K.Q. Weinberger, A.G. Wilson, GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration, in: Advances in Neural Information Processing Systems, 2018.
[87] L. Ardizzone, T. Bungert, F. Draxler, U. Köthe, J. Kruse, R. Schmier, P. Sorrenson, Framework for Easily Invertible Architectures (FrEIA), 2018-2022, URL https://github.com/vislearn/FrEIA.