

Article

Interpretable Calibration of Crystal Plasticity Model Using a Bayesian Surrogate-Assisted Genetic Algorithm

Shuaiyi Yang, Xuefeng Tang *, Lei Deng, Pan Gong , Mao Zhang, Junsong Jin and Xinyun Wang

State Key Laboratory of Materials Processing and Die and Mold Technology, School of Material Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

* Correspondence: xftang@hust.edu.cn

Abstract: The accurate calibration of material parameters in crystal plasticity models is essential for applying crystal plasticity (CP) simulations. Identifying these parameters usually requires unfeasible single-crystal experiments or expensive time costs due to the use of traditional genetic algorithm (GA) optimization. This study proposed an efficient and interpretable method for calibrating the constitutive parameters with macroscopic mechanical tests. This approach utilized the Bayesian neural network (BNN)-based surrogate-assisted GA (SGA) optimization method to identify a group of constitutive parameters that can reproduce the experimental stress–strain curve and crystallographic orientation by crystal plasticity simulation. The proposed approach was performed on the calibration of typical high-entropy alloy material parameters in two different CP models. The use of the surrogate model reduces the call count of simulation in the parameter searching process and speeds up the calibration significantly. With the help of infill sampling, the accuracy of this optimization method is consistent with the CP simulation and not limited by the accuracy of the surrogate model. Another merit of this method is that the pattern that the BNN surrogate found in the model parameters can be interpreted with its integrated gradients, which helps us to understand the relationship between constitutive parameters and the output mechanical response. The interpretation of BNN can guide further experiment design to decouple particular parameters and add constraints provided by the attached experiment or prior knowledge.



Citation: Yang, S.; Tang, X.; Deng, L.; Gong, P.; Zhang, M.; Jin, J.; Wang, X. Interpretable Calibration of Crystal Plasticity Model Using a Bayesian Surrogate-Assisted Genetic Algorithm. *Metals* **2023**, *13*, 166. <https://doi.org/10.3390/met13010166>

Academic Editor: Umberto Prisco

Received: 18 November 2022

Revised: 6 January 2023

Accepted: 7 January 2023

Published: 13 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: interpretable calibration; crystal plasticity; Bayesian surrogate; genetic algorithm; Bayesian neural network

1. Introduction

Under the impetus of digitalization and intellectualization tide of manufacturing, advanced simulation technology is required to accurately predict the manufacturing process and map the forming history to the product's performance [1]. The crystal plasticity (CP) model utilizes the knowledge gained from experimental and theoretical studies of deformation physics of single crystals and polycrystals to predict anisotropic deformation behavior in a natural way. These models give insight into the evolution of crystallographic texture, dislocation density, stored mechanical energy, deformation-driven athermal transformations (e.g., twinning), etc.

The constitutive laws in the CP model, including the phenomenological model and the physics-based model, involve a set of adjustable parameters to provide the physics of the material behavior. The model accuracy strongly depends on the selected value of the adjustable material parameters. Therefore, the identification of material parameters is an essential prerequisite for the high predictive ability of the CP model. Theoretically, most of the constitutive parameters can be obtained from the mechanical response of a single crystal. However, these experiments are often unfeasible as typical engineering materials are often available as polycrystals, and a special method is required to probe into the mechanical behavior of single grains. One approach often used for polycrystals is a nanoindentation

experiment. Another strategy involves the use of a coupled experimental–numerical procedure [2]. Since significant experimental effort is needed in these methods, identifying CP constitutive parameters via the inverse optimization methods from macroscopic stress–strain data determined by specific tensile or compression testing is generally preferred [3,4].

The procedure for the calibration of constitutive parameters with a macroscopic mechanical response is equivalent to solving the inverse problem of the CP model. The physic-based CP simulations, especially those of a full-field nature, often take minutes or hours to run [5]. Such optimization methods often require hundreds of simulations on different sets of parameters and take days or weeks to precisely calibrate the CP model. Thus, developing a computationally efficient and robust methodology to identify the constitutive parameters is significant for industry digitalization [6].

Many studies have employed gradient-based optimization methods for this purpose [4,7]. However, the main drawback of the gradient-based methods is that they are sensitive to the initial point, and the results are often inconsistent [8] since the constitutive laws are constructed with highly nonlinear equations, especially when encountering the micromechanics of complex engineering materials with multiple deformation mechanisms. Moreover, the gradient of the CP model is not always accessible. The reasons mentioned above limit the application of the gradient method in the parameter calibration of the CP model.

Another approach takes advantage of direct search methods, such as genetic algorithms (GAs) [9], searching the space by randomly generating solutions and providing new solutions based on the evaluation of the objective functions at the sampled points. As they negate the need for evaluating the gradient of the objective, direct search methods are able to optimize complex functions. These methods are capable of searching for the global optimum and do not have those drawbacks present in the gradient-based methods. However, these improvements come at the cost of computation efficiency [10].

The efficiency of GAs in CP parameter calibration is directly related to the call count of the CP simulation, which has the highest computational cost in fitness value calculation in the GA optimization process. Due to the expensive computational cost of numerical simulations using the CP model, it is rarely feasible to search a design space completely using simulation. The cost of the routine use of GAs can be cut down with the help of a surrogate model. So far, many studies have built surrogate models for the CP model to predict the physical response of a typical material, e.g., the texture evolution [11,12], dislocation density evolution [13], and stress distribution [14].

Data-based static models have been widely utilized to build the surrogate model for time-costing simulations in the optimization process [15,16]. In the research of Sedighiani et al. [6], a polynomial was fitted to approximate the constitutive response with the help of response surface methodology and using a GA to search the response surface in place of CP simulation. An artificial neural network (ANN)-based surrogate for CP simulation [17] is also a feasible approach to accelerating the process of GA searching. Data-based surrogate models, after training on a sufficient number of samples, can make an accurate prediction of the mechanical response much faster than their original counterpart. The surrogate-assisted GA (SGA) utilizes the surrogate model in fitness calculation and reduces the overall computational cost in the optimization procedure. Though some simulations are required to train the data-based surrogate model, the computational cost is still considered to be significantly lower than the cost of running the simulations required for GA searching.

However, the accuracy of the aforementioned surrogate-based method for CP parameter optimization is limited by the surrogate models. The optimization is completely based on the prediction of surrogate models after they are trained once on the initial samples provided by CP simulation. Additionally, this random initial sampling in the parameter space is redundant. It performs the same density of sampling at the promising point and around the optimum. In this study, a Bayesian neural network (BNN)-based surrogate is employed, and the infill sampling criteria (ISC) are utilized to allocate new sample points as well as to iteratively update the BNN surrogate during the optimization process. Moreover,

the pattern learned by the BNN surrogate is further studied, providing insight into the calibration process thanks to the integrated gradients method. The remainder of this article is organized as follows. The next section details a BNN SGA with Expected Improvement (EI) as the ISC. In Sections 3 and 4, we apply this methodology to calibrate the parameters of high-entropy alloys for phenomenological CP and a dislocation-density-based CP model with an experimentally determined mechanical response, based on which the impact of the ISC is discussed and the calibration result is analyzed with the information provided by the integrated gradients of the BNN surrogate.

2. Methodology

In this section, a novel approach is proposed to identify constitutive parameters of CP laws from macroscopic stress–strain curves using a BNN SGA. The construction of the BNN surrogate model is introduced, and the BNN SGA using EI as the infill sampling criteria in the acquisition process is detailed. The workflow of this method is presented in Figure 1. The experiment data are resampled and normalized to provide access to fitness for CP simulation and surrogate-predicted mechanical responses. The optimization started with a group of random initialized constitutive parameters, set as the initial population in the GA, which are optimized during iteration. After the surrogate model is trained on the data provided by CP simulation, the acquisition process is adopted to generate new solutions as well as assign a surrogate model for those solutions that are not expected to see improvement, and their fitness value is calculated with the surrogate model output in the next iteration. When the stop criteria of the optimization process are met, the best solution corresponding to the optimal constitutive parameters is output. The detailed optimization process of the BNN SGA is described in the following sections.

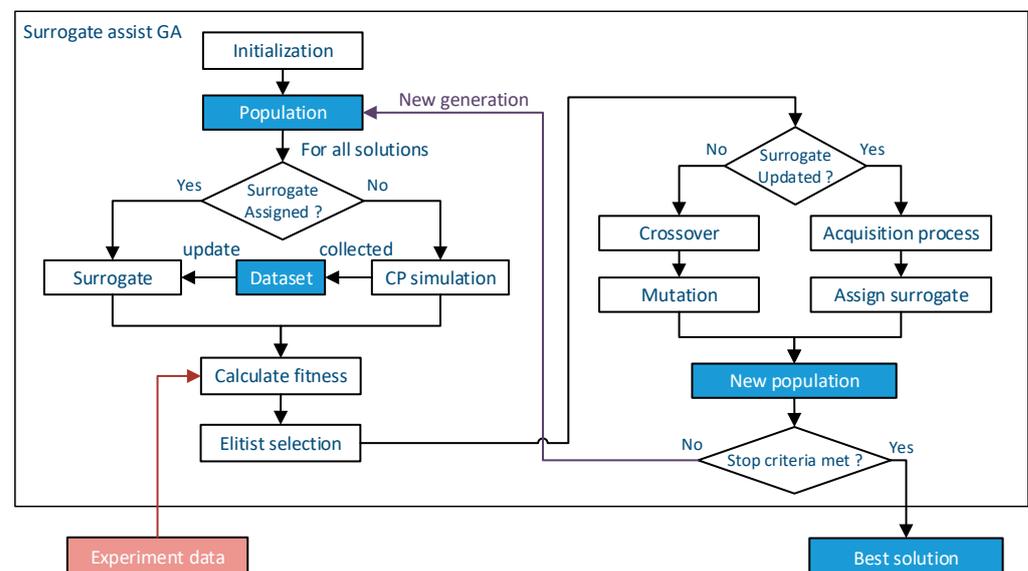


Figure 1. Workflow of CP parameter calibration using SGA.

2.1. Optimization Algorithm

GA is a powerful optimization method that is able to search for the global optimum rather than getting stuck in local optimums [18], and is thus considered an alternative for optimization with a nonlinear objective function. It is based on a randomized search technique that imitates the principles of natural selection and evolution processes. In this study, a GA is employed to optimize the input CP parameters until the difference between the simulation-predicted mechanical response and experimentally determined response is minimized. PyGAD v2.16.0 (University of Ottawa, Ottawa, ON, Canada) [19], an open-source Python library for GAs, is utilized and modified to perform the multi-processing surrogate-assisted GA optimization in our implementation.

A GA consists of a series of different steps. As shown in Figure 2, the first generation is generated by uniform sampling during initialization, which provides training data for the first update of the surrogate model. In the selection process, the fitness value is calculated for each solution, based on which the parents are selected to produce the next generation. The genes in the offspring solutions are then modified with mutation and crossover operators. In this study, the mutation is implemented by randomly adding value to the genes, and the crossover operator swaps the genes of the parents after the crossover point. The selection, crossover, and mutation processes are iteratively performed on the new generation until the global optimum is found.

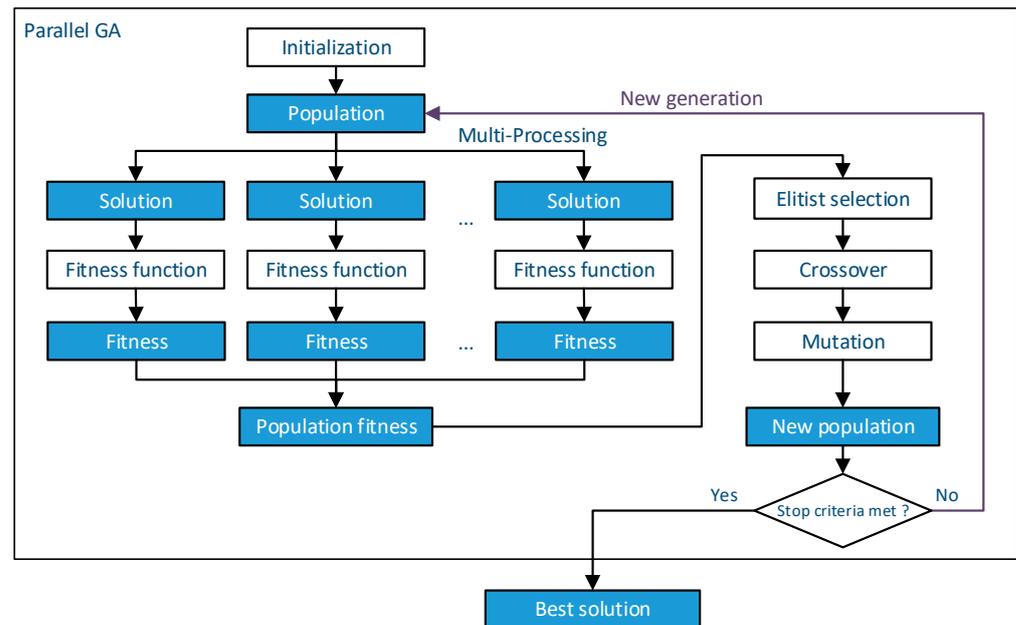


Figure 2. Workflow of parallel GA.

The experimentally measured stress–strain curve is set as the target response, and an objective function is introduced based on the stress–strain curve. The root mean squared error (RMSE) is used to measure the difference between experimental and simulated results at the selected sample points, and is set as the objective function of the optimization:

$$d^l(\theta) = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{\hat{\sigma}_i(\theta) - \sigma_i}{\sigma_{\max}} \right)^2} \quad (1)$$

where σ_i is the experimentally measured stress at the i th strain sample point, σ_{\max} is the maximum value of the measured stress, $\hat{\sigma}_i(\theta)$ is the predicted stress from the BNN surrogate or CP simulation for a group of constitutive parameters θ at the i th strain sample point, N is the number of sample points, and $d^l(\theta)$ denotes the relative difference between two curves under load condition l for the parameters. The reciprocal of the objective function is used to calculate the final fitness of solutions:

$$\text{fitness} = \frac{1}{d^l(\theta)} \quad (2)$$

In elitist selection process, a few of the best solutions with the highest fitness are preserved and passed on to the next generation without modifying their genes. Elitist selection maintains the elite solutions and guarantees that achievement in fitness can be passed from generation to generation. The parameters for GA optimization are listed in Table 1.

Table 1. The parameters for GA optimization.

Parameter	Value
Population size	32
Number of parents	16
Number of elitists	4
Mutation rate	0.1
Mutation type	Adding random value
Crossover rate	0.2
Crossover type	Single-point crossover
Number of parallel processing	12

The calculation of fitness can be implemented as parallel processing since the solutions are independent of each other during that process. However, in most GA applications, parallel processing does not reduce the computational cost compared to regular processing since multi-processing usually has a heavy computational cost. If the cost of assigning multi-processing is negligible compared to that of the fitness calculation, the overall run time can be significantly reduced by using multi-processing. In the context of CP model calibration, CP simulation always has a much higher computational cost than creating multi-processing. The calibration cost could then be slashed by using multi-processing.

2.2. BNN-Based Surrogate Model

Surrogate models work as a replacement for the expensive simulation in optimization. Previous studies suggested using an ANN and its variants to build the surrogate for the CP model [11,12]. Once sufficiently trained on the samples, the ANNs can obtain high accuracy in prediction, and this process runs very fast owing to the efficient matrix operation. Moreover, ANNs allow for incremental learning, and the model can be refined after being trained on new samples during optimization.

ANNs are proposed as a tool for classification and regression and do not provide uncertainty information around their predictions [20]. Since EI computation requires uncertainty information, coupling an ordinary ANN in the SGA method with EI as infill sampling criteria is not feasible. Meanwhile, the BNN could provide the posterior probability distribution $p(\mathbf{W}|\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}})$ over the weights [21], rather than the direct value of weights, and thus provides the uncertainty for EI calculation. Though using the traditional BNN is rarely feasible due to its expensive computational cost in training, this can be solved by using the MCDropout-based BNN.

The MCDropout method allows efficient BNN training by applying the Dropout technique in the training for the original ANN. Dropout is a regularization technique used in ANN training to reduce the flexibility of ANNs and prevent them from overfitting [22]. It is proven that adding dropout layers before the weight layer in a neural network is mathematically an approximation to the deep Gaussian process [20] i.e., the BNN. For the MCDropout-based BNN, the distribution of predicted parameters can be accessed with Monte Carlo (MC) estimation, i.e., performing stochastic forward propagation through the network for certain times and then calculating the average and variance of the prediction.

In this study, the MCDropout-based BNN is utilized as the surrogate model to predict the stress–strain curve for the input constitutive parameters. The BNN provides the distribution of the predicted objective value to calculate EI in the acquisition process. The neural network is first trained on the sample provided by CP simulation in the initial generation and then updated on the sample points with the highest EI and assigned to perform CP simulation in subsequent generations' acquisition process.

2.2.1. Hyperparameters and Structure of BNN

The BNN utilized in this study consists of two parts, one from the input layer to output layer 1, and another from output layer 1 to output layer 2. The BNN is trained stepwise, instead of directly predicting the objective. It is trained to first predict the stress–strain curve

in the form of feature parameters, and then to predict the objective from the curve features (Figure 3). This middle layer makes the output feature accessible for the surrogate model and allows for the extraction of BNN learned pattern. The first part, denoted as BNN₁, is trained to capture the relationship between the stress–strain curve and the CP parameters, and works as the surrogate for CP simulation. The latter part, denoted as BNN₂, predicts the objective $d^k(\theta)$ distribution with the help of MCDropout. Both parts of the network have three dense layers, making them capable of capturing nonlinear relationships. The major hyperparameters and structure of the BNN are listed in Table 2.

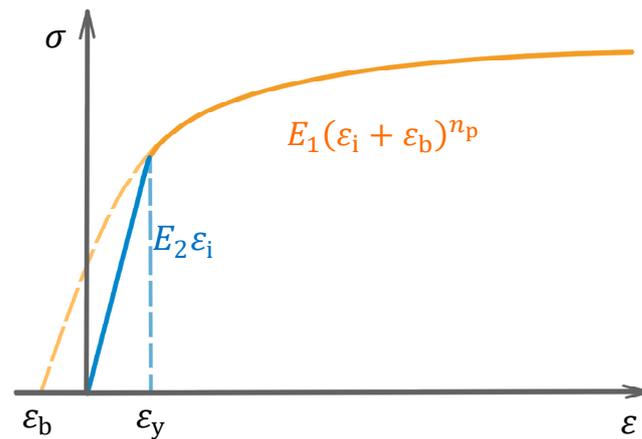


Figure 3. Linear-elastic exponential-plastic model.

Table 2. Hyperparameters and structure of BNN.

Layers	Cell Number	Activation	Notes
Input layer	-	-	Normalized CP parameter $\bar{\theta}$
Dense layer 1	32	ReLU	-
Dense layer 2	16	ReLU	-
Output layer 1	-	Sigmoid	Output the curve features
Dense layer 3	16	ReLU	-
Dropout layer 1	16	-	Dropout rate = 0.2
Dense layer 4	8	ReLU	-
Dropout layer 2	8	-	Dropout rate = 0.2
Output layer 2	1	Sigmoid	Output the objective $d^l(\theta)$

Activation functions transform the output of weight, typically nonlinearly, enabling the network to learn nonlinear functions. ReLU stands for the Rectified Linear Units, which is introduced in [23]. The dropout rate controls the percent of closed cells in the forward propagation. It should be mentioned that the BNN inputs (i.e., the CP parameters) have been normalized, and the output stress–strain curve is represented by a certain number of features. These transformations on the data extend the BNN to fit the data with different scales and units.

2.2.2. Data Processing

The activation for the output layer of BNN is the sigmoid function, which has a range of (0,1). Hence, the output must be rescaled or transformed to reproduce the physical quantities. Though there are no exact bounds for the input data, rescaling is performed on the input to speed up and stabilize the training process. If the BNN input vector has various scales in different dimensions, the gradient descent method for BNN training cannot effectively optimize the parameters in the BNN. In typical BNN applications, the data are normalized before training, or batch normalization (BN) [24] is added before the activation in the first layer.

In this study, input CP parameters were rescaled before being fed into the input layer, and the features of the stress–strain curve were extracted as the training target for output layer 1. The objective value was $d^l(\theta)$, which has already been normalized and can be directly set as the target for output layer 2. To normalize the CP parameters with different units, the CP parameters were rescaled into a range of (0, 1) using the min–max scaler:

$$\bar{\theta}_i = \frac{\theta_i - \theta_{i \min}}{\theta_{i \max} - \theta_{i \min}} \quad (3)$$

where $\bar{\theta}_i$ is the l th normalized CP parameter, θ_i is the i th original CP parameter, and $\theta_{i \min}$ and $\theta_{i \max}$ are the manually estimated lower and upper bound for the i th CP parameter, respectively. The normalized CP parameter $\bar{\theta}$ is utilized as the input for BNN₁.

The stress–strain value (σ, ε) is a high-dimensional datum. It would be unfeasible for the BNN-based surrogate model to predict the stress at a massive number of sample points with a few input parameters. Simply reducing the number of sample points can lessen the difficulty of training the surrogate model, but it may fail to capture the features of the stress–strain curve. To overcome the “dimension curse”, the dimension of stress–strain curve data should be reduced before feeding it into the surrogate model. As shown in Figure 3, a prior model, the linear-elastic exponential-plastic model f_p , is used to describe the stress–strain curve:

$$\sigma_i = f_p(\varepsilon_i, E_1, \varepsilon_b, \varepsilon_y, n_p) = \begin{cases} E_1(\varepsilon_i + \varepsilon_b)^{n_p} & (\varepsilon_i \geq \varepsilon_y) \\ E_2\varepsilon_i & (\varepsilon_i < \varepsilon_y) \\ E_2\varepsilon_y = E_1(\varepsilon_b + \varepsilon_y)^{n_p} & \end{cases} \quad (4)$$

where σ_i, ε_i is the i th stress and strain sample value, respectively, ε_y is the yield strain, ε_b is the bias for the exponential part, and n_p is the exponent, E_1 is the variable controlling the value of yield stress, and E_2 is the elastic modulus. The parameter E_2 can be solved thanks to the continuity of the curve, and there are four independent feature parameters ($E_1, \varepsilon_b, \varepsilon_y, n_p$) describing the stress–strain curve.

The feature parameters ($E_1, \varepsilon_b, \varepsilon_y, n_p$) have different scales and cannot be directly fitted by the BNN. To enable the BNN to predict feature parameters, the stress–strain should first be normalized before fitting the linear-elastic exponential-plastic model:

$$\bar{\sigma} = \frac{\sigma}{\sigma_{max}} \quad (5)$$

$$\bar{\varepsilon} = \frac{\varepsilon}{\varepsilon_{max}} \quad (6)$$

where $\bar{\sigma}, \bar{\varepsilon}$ are the normalized values of σ, ε , respectively, which have a range of (0, 1). $\sigma_{max}, \varepsilon_{max}$ are the maximum stress and strain value, respectively, of the experimental measured stress–strain curve. The normalized stress and strain values are used to calculate the normalized feature parameters of the linear-elastic exponential-plastic model f_p^{norm} .

$$\bar{\sigma}_i = f_p^{norm}(\bar{\varepsilon}_i, \mathbf{k}) = \begin{cases} k_2(\bar{\varepsilon}_i + k_1)^{k_3} & (\bar{\varepsilon}_i \geq k_4) \\ \frac{k_2}{k_4}(k_4 + k_1)^{k_3}\bar{\varepsilon}_i & (\bar{\varepsilon}_i < k_4) \end{cases} \quad (7)$$

where k_1, k_2, k_3, k_4 are the normalized curve features, whose range falls within (0, 1). These feature parameters can be calculated by the gradient method or stochastic optimization, e.g., difference evolution and the Newton-CG algorithm. The mean squared error is adopted as the objective for the four feature parameters in the linear-elastic exponential-plastic model to fit the stress–strain curve:

$$\mathbf{k} = \arg \min_k \left\{ \text{MSE} \left(\bar{\sigma} - f_p^{norm}(\bar{\varepsilon}, \mathbf{k}) \right) \right\} \quad (8)$$

As shown by the calibration result in Section 4, the linear-elastic exponential-plastic model is flexible enough to fit a variety of stress–strain curves obtained from experiments and simulations with high accuracy. Moreover, the computational cost of fitting the linear-elastic exponential-plastic model is negligible compared to that of CP simulation.

The BNN₁ is trained to correlate the parameter of the crystal plasticity model and these four featured parameters rather than the discrete stress–strain points. It reduces the stress–strain data from high-dimensional data (one sample point corresponds to one dimension) to several featured parameters ranging from 0 to 1. This is beneficial for the training of BNN. In addition, the analytical equation helps us to interpret the relationship between the mechanical response and parameters of the CP model by using integrated gradients. It gives specific meanings for the output of the BNN, which enables the interpretation of the calibration.

2.2.3. Implementation and Training of the Surrogate Model

The workflow of the BNN surrogate is presented in Figure 4. In the process of prediction, the first part of the BNN accepts normalized parameters as input and outputs the curve feature vector:

$$\hat{y}_1 = k_{\text{pred}} = \text{BNN}_1(x_1) \tag{9}$$

where $x_1 = \bar{\theta}$ is the input for BNN₁, k_{pred} is the feature vector predicted by BNN₁, equals to \hat{y}_1 , the output of BNN₁.

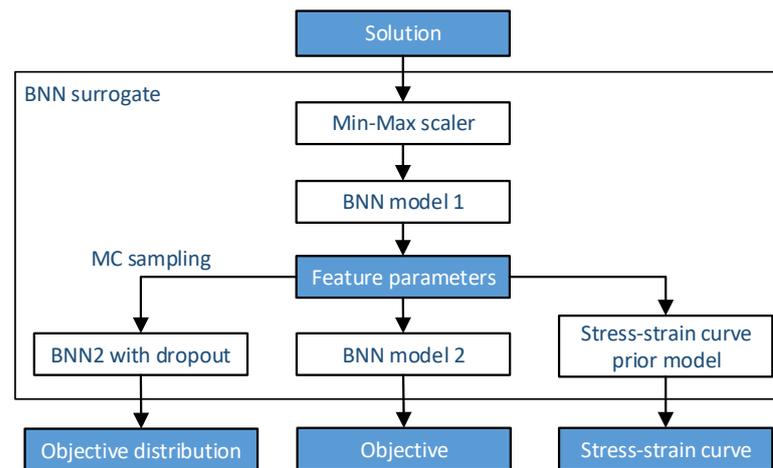


Figure 4. Schematic diagram of BNN surrogate model.

The target feature parameters y_1 used for training BNN₁ are calculated from the simulation output stress by optimizing feature parameters:

$$y_1 = k_{\text{sim}} = \arg \min_k \left\{ \text{MSE} \left(\bar{\sigma}_{\text{sim}} - f_p^{\text{norm}}(\bar{\epsilon}, k) \right) \right\} \tag{10}$$

where $\bar{\sigma}_{\text{sim}}$ is the normalized computational von Mises stress, and k_{sim} is the feature vector calculated from the simulation result.

For the second part of the BNN, the predicted feature parameters \hat{y}_1 are accepted as input, and the RSME of the curve features defined in Equation (1) is set as the target output for BNN₂. The BNN₂ is trained with Dropout method, whose predict value and confidence is obtained by MC estimation.

$$\hat{y}_2(x_2) = E[\text{BNN}_2(x_2)] \tag{11}$$

$$\hat{s}_2(x_2) = \sqrt{D[\text{BNN}_2(x_2)]} \tag{12}$$

where $x_2 = \hat{y}_1$ is the input for BNN₂, \hat{y}_2 is the objective predicted by BNN₂, and \hat{s}_2 is the standard deviation around the prediction. E and D represent the expectation and variance estimation, respectively. The Monte Carlo sample number N_{MC} is set to 40 in this study.

The target objective y_2 for training BNN₂ is calculated from the simulation output stress by:

$$y_2 = d_{sim}^l(\theta) \quad (13)$$

The training samples (x_1, y_1) and (x_2, y_2) provided by simulation are used to train the BNN surrogate by minimizing the loss between y_1, \hat{y}_1 and y_2, \hat{y}_2 . The mean square error (MSE) between the prediction and target value is chosen as the loss function for both BNN₁ and BNN₂,

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (14)$$

where n is the number of output features, and \hat{y}_i and y_i are the BNN output and target value, respectively. The BNN cost for the current batch of training data is defined as the mean loss of all the samples:

$$\mathcal{J} = \frac{1}{m} \sum_{i=1}^m \mathcal{L}_{MSE} \quad (15)$$

where \mathcal{J} is the cost function, and m is the size of the training batch. The BNN can be incrementally updated with a new batch of data. Incremental learning negates the need for repeated training of BNN on all the samples, significantly reducing the training time. Using a batch to represent all the samples and training on it could cause gradient instability. This problem can be reduced by tuning the batch size and training epochs.

Adaptive Moment Estimation (Adam) [25] is utilized to optimize the training parameters of the BNN. The training parameters are listed in Table 3.

Table 3. The hyperparameters for BNN training.

Hyperparameters	Value
Batch size	16
Training epochs	40
Learning rate	0.002
Momentum decay β_1	0.9
Scaling decay β_2	0.999

Another data-based surrogate utilized in this study is the Lookup-based surrogate, which looks up for the similar input CP parameters from the simulated solutions' dataset and directly returns their result. This strategy reduces the computational cost of calculating fitness on similar solutions. In GA searching, similar solutions can occur many times, especially when the population's gene converges to the optimum. Thus, the utilization of the Lookup surrogate can effectively reduce the computational cost of the simulation for similar solutions. The threshold for adopting the Lookup surrogate is defined as follows:

$$d(\bar{\theta}_c, \bar{\theta}_s) = \|\bar{\theta}_c - \bar{\theta}_s\| \leq d_{th} \quad (16)$$

where d_{th} is the threshold for performing the Lookup surrogate model, $\bar{\theta}_c$ and $\bar{\theta}_s$ are the normalization of current candidate CP parameters θ_c and simulated CP parameters θ_s , respectively, $d(\bar{\theta}_c, \bar{\theta}_s)$ denotes the distance between $\bar{\theta}_c$ and $\bar{\theta}_s$, and the L_2 -norm $\|\cdot\|$ is used to measure the distance.

The Lookup surrogate is adopted when a group of parameters is firstly assigned to be simulated and the similar parameters group (i.e., the distance is less than d_{th}) has already been simulated. In this study, the d_{th} is set to 0.001 and can be adjusted to a smaller value to obtain higher accuracy near the end of GA searching. It should be noted that the result of the Lookup surrogate will not be stored to train the BNN, since these samples have already been learned by the BNN.

2.3. MCDropout-Based BNN-Assisted GA

Many surrogate-based optimization methods have been utilized in recent studies to calibrate the CP parameters [6,17,26]. These surrogate models are only trained on initial samples, and optimization is performed based on the response of the surrogate model. In surrogate-based optimization, adjustable parameters would not converge to the global optimum if the surrogate model is built from only the initially sampled data or is simply an updated model with the current optimum as the new sample point [27]. This is because the surrogate model may not be sufficiently trained at the sample point in the optimization and is not accurate enough regarding its proximity to the global optimum. An iterative sampling refinement method is utilized to solve this problem, which allows the surrogate model to be updated on the new sample points generated during the optimization process. The criteria used to search the parameter space and refine the surrogate model with a promising sample point are the so-called infill sampling criteria [28]. Among various ISC, maximizing the EI is a widespread criterion and performs consistently well in the instances considered in the study compared with other ISC [29].

In this study, iterative sampling refinement is utilized so that the surrogate-assisted optimization can reach the same accuracy as it could be achieved by directly minimizing the original objective function (i.e., using the CP simulation physical response to calculate the fitness value in a GA). Maximizing EI is chosen as the infill sampling criterion in the acquisition process, which generates a new population after the surrogate is trained [30]. In the acquisition process, the second part of the BNN is utilized to predict the objective distribution for the EI calculation.

2.3.1. Infill Sampling Criteria

The iterative sampling refinement method uses new sample points in promising regions to refine the surrogate model. The ISC are used to determine whether to sample the parameter space and refine the surrogate model or utilize the surrogate output, which is one of the key techniques in surrogate-based optimization. The EI for the candidate solution x is given by:

$$EI(x) = (y_{min} - \hat{y}(x))\Phi\left(\frac{y_{min} - \hat{y}(x)}{\hat{s}(x)}\right) + \hat{s}(x)\phi\left(\frac{y_{min} - \hat{y}(x)}{\hat{s}(x)}\right) \quad (\hat{s} > 0) \quad (17)$$

where Φ is the cumulative distribution function for the normal law and ϕ is its probability density function, y_{min} is the best simulated cost found so far, $\hat{y}(x)$ is the predicted cost provided by the surrogate, and $\hat{s}(x)$ is the variance provided by the surrogate model for its predicted cost. In this study, the EI is calculated based on the BNN-predicted objective and its uncertainty according to Equations (11) and (12).

2.3.2. Acquisition Process

Once the surrogate is trained on the data provided by the initial generation, the acquisition process is activated. Figure 5 shows the workflow of the acquisition process. The acquisition process generates and selects sample points to perform CP simulation from the candidate solutions according to the ISC, and is employed as a trade-off between exploration and exploitation [31]. For the selected parent solutions with higher fitness values in the population, their offspring are generated by crossover and mutation. Then, the BNN surrogate is utilized to predict the objective with uncertainty information for all the offspring solutions. Their EI is computed by Equation (17). A proportion of children with the greater EI value are assigned to perform CP simulation. The remaining children who are less promising are set to be predicted by the surrogate model to gain speedup. In this study, the proportion assigned to the BNN surrogate model p_{BNN} is set to different values, and its effect will be discussed.

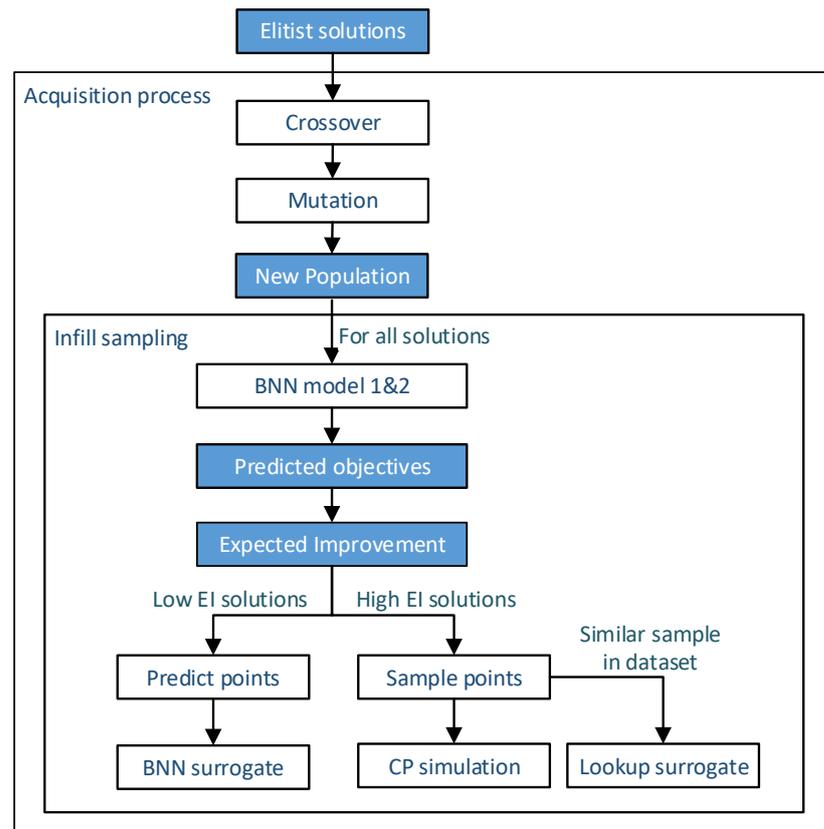


Figure 5. Workflow of acquisition process and infill sampling.

3. CP Model and Set-Up of SGA

This section employs the proposed BNN SGA optimization method to calibrate the phenomenological CP model and dislocation-density-based CP model of a CoCrFeMnNi high-entropy alloy (HEA). For this HEA, a twelve-slip system of the $\{111\} \langle 110 \rangle$ -type was considered, and the deformation twinning was omitted. The CP simulation toolkit DAMASK v3.0.0 alpha4 (Max-Planck-Institut für Eisenforschung GmbH, Düsseldorf, Germany) [32] was employed to perform the simulations. A cubic representative volume element (RVE) with 100 grains was used to reproduce the experimental result, as shown in Figure 6. The average grain size is about $6 \mu\text{m}$, and uniaxial compression along the x -axis with a strain rate of 0.001 s^{-1} was applied. Python v3.8.10, NumPy v1.21.2, SciPy v1.7.1, and TensorFlow v2.8.0 (Google, Mountain View, CA, USA) were employed to construct the BNN SGA.



Figure 6. The representative volume element (RVE) used for simulation.

3.1. CP Model for Calibration

3.1.1. Phenomenological CP Model

The applied phenomenological CP model, which was first proposed by Hutchinson for FCC crystals, is the most widely used phenomenological model [33]. The evolution of the shear rate ($\dot{\gamma}^\alpha$) on each slip system is given by:

$$\dot{\gamma}^\alpha = \dot{\gamma}_0^\alpha \left| \frac{\tau^\alpha}{\zeta^\alpha} \right| \text{sgn}(\tau^\alpha) \quad (18)$$

where $\dot{\gamma}_0^\alpha$, τ^α , and ζ^α are the reference shear rate, resolve shear stress, and slip resistance of slip system α , respectively.

The slip resistance of slip system α , ζ^α , increasingly evolves from the initial value ζ_0 to a saturation value ζ_∞^α governed by the shear on slip and twin systems:

$$\dot{\zeta}^\alpha = h_0^{s-s} \sum_{\alpha'=1}^{N_s} \left| \dot{\gamma}^{\alpha'} \right| \left| 1 - \frac{\zeta^{\alpha'}}{\zeta_\infty^{\alpha'}} \right|^\alpha \text{sgn} \left(1 - \frac{\zeta^{\alpha'}}{\zeta_\infty^{\alpha'}} \right) h^{\alpha\alpha'} \quad (19)$$

Here, h_0^{s-s} , a are the hardening parameters. N_s is the total number of slip systems. $h^{\alpha\alpha'}$ is the interaction coefficient between slip system α , and α' is assumed to be 1.0 for coplanar slip systems and 1.4 for non-coplanar systems.

3.1.2. Dislocation-Density-Based CP Model

The dislocation-density-based CP model is the built-in model of DAMASK. A detailed description can be found in Ref. [32]. The shear rate is given by the Orowan equation as

$$\dot{\gamma} = \rho b_s v_0 \exp \left[-\frac{Q_s}{k_B T} \left\{ 1 - \left(\frac{\tau_{\text{eff}}}{\tau_{\text{sol}}} \right)^{p_s} \right\}^{q_s} \right] \text{sgn}(\tau) \quad (20)$$

where b_s is the Burgers vector length of slip families, v_0 is the initial glide velocity, Q_s is the activation energy of dislocation glide, k_B is the Boltzmann constant, T is the temperature, τ_{eff} is the effective shear stress, and τ_{sol} is the solid solution strength. p_s and q_s are the fitting parameters for glide velocity. The effective shear stress τ_{eff} is computed as:

$$\tau_{\text{eff}} = \begin{cases} |\tau| - \tau_{\text{pass}} & \text{for } |\tau| > \tau_{\text{pass}} \\ 0 & \text{for } |\tau| \leq \tau_{\text{pass}} \end{cases} \quad (21)$$

and the pass stress τ_{pass} is given by:

$$\tau_{\text{pass}}^\alpha = G b_s^\alpha \left(\sum_{\alpha'} h^{\alpha\alpha'} (\rho^{\alpha'} + \rho_{\text{dip}}^{\alpha'}) \right)^{0.5} \quad (22)$$

where G is the shear modulus, and $h^{\alpha\alpha'}$ are the slip–slip interaction matrices.

The evolution of dislocation densities is determined by dislocation multiplication, annihilation, and dipole formation. The evolution of dislocation density is given by:

$$\dot{\rho} = \frac{|\dot{\gamma}|}{b_s \Lambda_s} - \frac{2\bar{d}}{b_s} \rho |\dot{\gamma}| \quad (23)$$

$$\dot{\rho}_{\text{dip}} = \frac{2(\bar{d}-\bar{d})}{b_s} \rho |\dot{\gamma}| - \frac{2\bar{d}}{b_s} \rho_{\text{dip}} |\dot{\gamma}| - \rho_{\text{dip}} \frac{4v_{\text{cl}}}{\bar{d}-\bar{d}} \quad (24)$$

where Λ represents the mean free path (MFP) for dislocation, which governs the dislocation storage process. v_{cl} is the dislocation climb velocity and is given by:

$$v_{\text{cl}} = \frac{G D_0 v_{\text{cl}}}{\pi(1-\nu)k_B T} \frac{1}{\bar{d}+\bar{d}} \exp \left(-\frac{Q_{\text{cl}}}{k_B T} \right) \quad (25)$$

where \hat{d} denotes the glide plane distance below which two dislocations form a stable dipole and \check{d} denotes the glide plane distance below which two dislocations are annihilated. These are defined as:

$$\hat{d} = \frac{3Gb_s}{16\pi|\tau|} \quad (26)$$

$$\check{d} = D_a b_s \quad (27)$$

where D_a is a fitting parameter.

The MFP describes the pileup of dislocations in front of grain boundaries, dislocation–dislocation interaction, and the formation of twins, and is composed as:

$$\frac{1}{\Lambda_s} = \frac{1}{D} + \frac{1}{\lambda_s} \quad (28)$$

$$\frac{1}{\lambda_s} = \frac{1}{i_s} \left(\sum_{\alpha'=1}^{N_s} p^{\alpha\alpha'} \left(\rho^{\alpha'} + \rho_{\text{dip}}^{\alpha'} \right) \right)^{0.5} \quad (29)$$

where D is the average grain size, i_s is a fitting parameter, $\rho^{\alpha\alpha'}$ are projections for the forest dislocation density. Deformation twinning is omitted here since it infrequently occurred in this HEA.

3.2. Set-Up of SGA

3.2.1. Experiment Data

The experimental stress–strain response is obtained by a uniaxial compression at a strain rate of 10^{-3}s^{-1} . The experimentally measured stress–strain curve is set as the target to be reproduced by a group of CP parameters. The initial orientation of the grains in the simulation is generated according to the orientation distribution of the texture. Figure 7 illustrates the pole figures characterizing the measured initial orientation and the sampled initial orientation used for CP simulation. As the initial orientations of the nodes within one grain are the same, the initial orientation distribution for CP simulation is very close to the experimental result. No evident texture was observed in the two pole figures, and the intensity of the initial orientation for CP simulation is slightly higher than that of the experimental one due to the smaller number of sampling grains of the RVE. The final orientation is utilized to validate the calibration result obtained from fitting the stress–strain curve. In this study, MTEX, the MATLAB toolbox for crystallographic textures analysis, is employed to process the EBSD data and draw the pole figures for material textures.

3.2.2. Input Parameters for SGA

The elastic constants C_{11} , C_{12} , C_{44} , reference shear rate $\dot{\gamma}_0$, and the magnitude of the Burgers vector were regarded as the definitive constitutive parameters for CP simulation, as listed in Table 4. Parameters p_s and q_s in the dislocation-density-based CP model can be set to 1.0 for cuboidal shapes of local obstacles [34]. All the simulations were conducted at room temperature, and the temperature T was thus set as 300 K. The adjustable material parameters of the phenomenological and dislocation-density-based CP model and their reference ranges, which need to be optimized to reproduce the experimental stress–strain curves and pole figures, are listed in Tables 5 and 6, respectively.

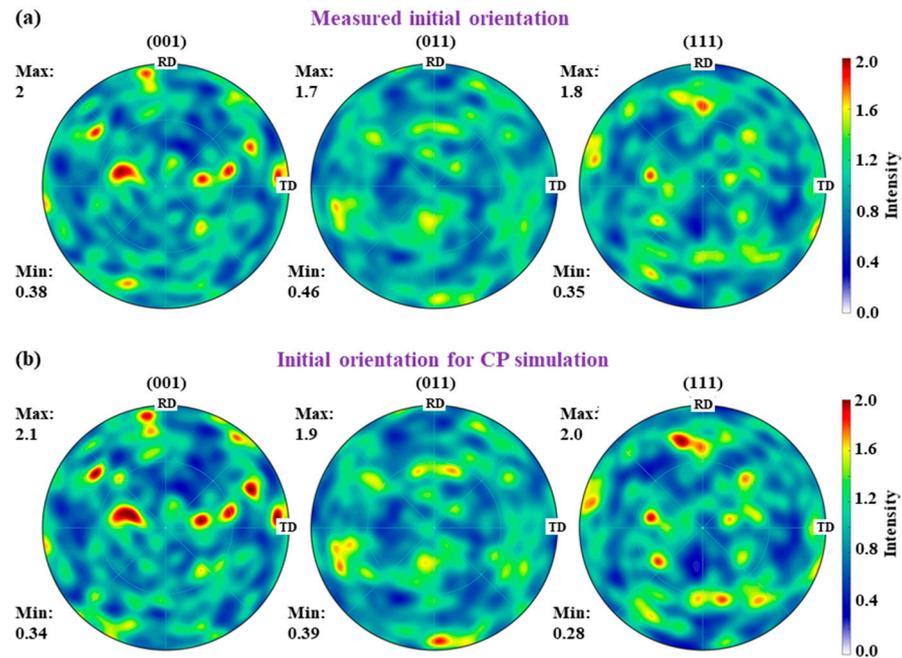


Figure 7. Pole figure characterizing (a) the measured initial orientation and (b) the initial orientation for CP simulation.

Table 4. Definitive constitutive parameters for the phenomenological CP model.

Variable	Units	Value	Note
C_{11}	GPa	245.1	Elastic constant [35]
C_{12}	GPa	148.9	Elastic constant [35]
C_{44}	GPa	191.5	Elastic constant [35]
$\dot{\gamma}_0$	s^{-1}	0.001	Reference shear rate [32]
b_s	m	2.54×10^{-10}	Burgers vector magnitude of slip system
p_s	-	1.0	p -exponent in glide velocity [34,36]
q_s	-	1.0	q -exponent in glide velocity [34,36]

Table 5. Ranges of the adjustable parameters for the phenomenological CP model.

Variable	Units	Range	Note
ζ_0	MPa	[50, 500]	Slip resistance
ζ_∞	MPa	[500, 2500]	Saturation stress
h_0	MPa	[100, 700]	Slip-hardening parameter
n	-	[5, 30]	Strain rate sensitivity parameter
a	-	[1, 5]	Slip-hardening parameter

3.2.3. Hyperparameters for SGA

Table 7 lists the hyperparameters for surrogate-assisted searching that control the use of Lookup and BNN surrogates. The proportion of the performing BNN surrogate p_{BNN} is set to 0.2, 0.4, 0.6, and 0.8 in different runs, the results of which are compared and discussed in Section 4.

Table 6. Ranges of the adjustable parameters for the dislocation-density-based CP model.

Variable	Units	Range	Note
ρ_0	m^{-2}	$[5 \times 10^{10}, 5 \times 10^{11}]$	Initial dislocation density
$\rho_{\text{dip}0}$	m^{-2}	$[0.1, 0.8]$	Initial dipole dislocation density
v_0	$\text{m} \cdot \text{s}^{-1}$	$[1000, 10000]$	Initial glide velocity
Q_s	J	$[5 \times 10^{-20}, 5 \times 10^{-19}]$	Activation energy for dislocation glide
Q_{cl}	J	$[5 \times 10^{-20}, 5 \times 10^{-19}]$	Activation energy for climb
τ_0	MPa	$[5.0, 200.0]$	Solid solution strength
i_s	-	$[30.0, 100.0]$	Controlling dislocation mean free path
D_0	$\text{m}^2 \cdot \text{s}^{-1}$	$[1 \times 10^{-6}, 3 \times 10^{-5}]$	Vacancy diffusion prefactor
D_a	-	$[3.0, 15.0]$	Minimum dipole distance

Table 7. Hyperparameters for SGA.

Hyperparameters	Value
Threshold for Lookup surrogate d_{th}	0.001
Number of MCDropout samples N_{MC}	40
Proportion of BNN surrogate p_{BNN}	0.2, 0.4, 0.6, 0.8

4. Results and Discussion

4.1. Optimization Process

According to the input parameters and the SGA hyperparameters listed in Tables 4–7, four SGA searches are performed with different p_{BNN} values. In this section, we take the phenomenological CP model as an example, and the influence of p_{BNN} on optimization and the speedup provided by the SGA are discussed by analyzing the data extracted from their optimization processes.

4.1.1. Influence of p_{BNN}

Figure 8 shows the best fitness in each generation calculated by phenomenological CP simulation and BNN surrogate model under various p_{BNN} . As shown in Figure 8, p_{BNN} affects the fitness improvement of the solutions assigned to CP simulation. With the increase in p_{BNN} , the best fitness values calculated by the BNN surrogate and CP simulation tended to mix with each other, implying a worse performance of the acquisition process. This can be explained by two related aspects. Firstly, the BNN surrogate relies on the samples provided by CP simulation to be incrementally trained, improving its accuracy. On the other hand, the ISC rely on the BNN's accuracy to provide more confidence in assigning CP simulation to those solutions that are expected to gain higher fitness. A higher p_{BNN} value leads to a lack of training data for the BNN surrogate and a lack of confidence in the ISC for model assignment in the acquisition process and thus the failure to allocate promising sample points. A lower p_{BNN} value guarantees the BNN surrogate's accuracy and the stability of the acquisition process, while losing the speedup provided by using the surrogate model. To settle this contradiction, we defined a cumulative fitness improvement (CFI) per CP simulation to measure the effectiveness of the SGA process:

$$CFI(n) = \sum_{i=1}^n \frac{fitness_{\text{best}}(i) - fitness_{\text{best}}(i-1)}{N_{CP}(i)} \quad (30)$$

where $fitness_{\text{best}}(i)$ is the best fitness found till generation i , $N_{CP}(i)$ denotes the number of CP simulations performed in generation i , and n is the current generation number.

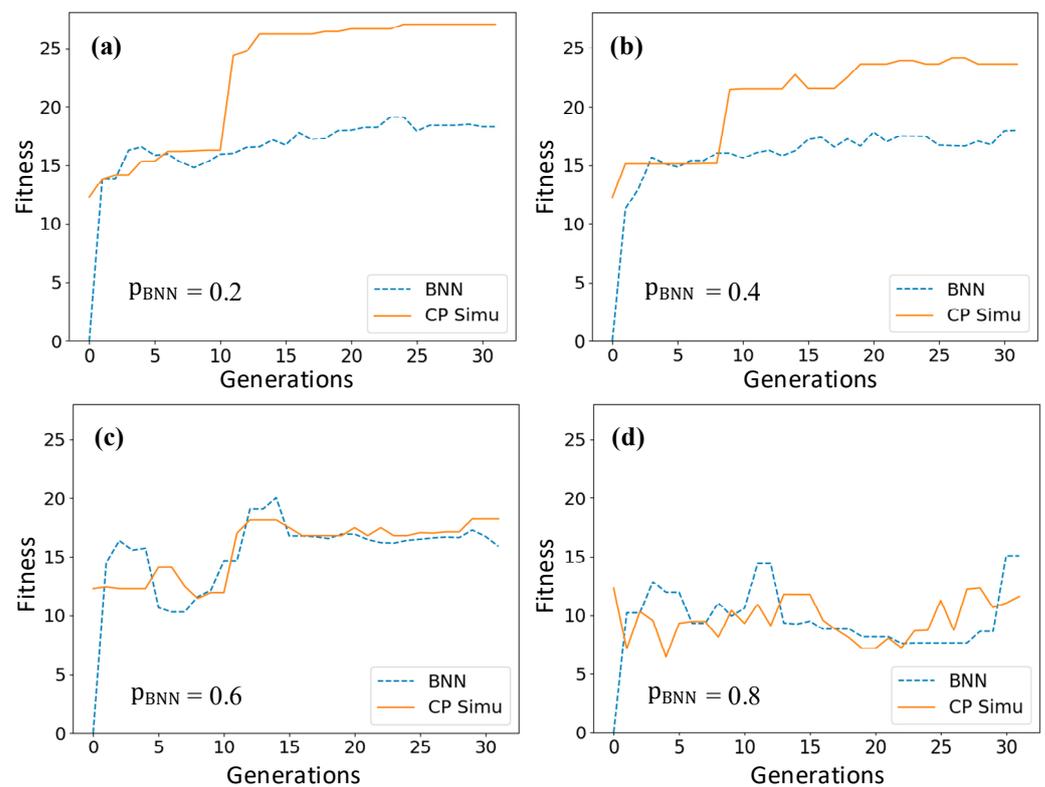


Figure 8. The best fitness in each generation calculated by phenomenological CP simulation and BNN when p_{BNN} equals (a) 0.2, (b) 0.4, (c) 0.6, and (d) 0.8.

CFI generally describes the fitness improvement contributed by each CP simulation during the SGA optimization process. The CFI in different SGA runs for the phenomenological CP model under different p_{BNN} is presented in Figure 9, where the $p_{BNN} = 0.4$ case shows a consistently higher CFI. The CFI progressively increases when $p_{BNN} = 0.2$, indicating a stable searching process but with more computational cost. When $p_{BNN} = 0.6$, the CFI is slowly increasing with fluctuations, indicating the acquisition process started to lose efficacy. It is worth noting that when $p_{BNN} = 0.8$, due to the interference of the surrogate model, the SGA search has not made progress in fitness improvement, resulting in a negative value. According to the discussion above, $p_{BNN} = 0.4$ is considered the best value for obtaining a balance between the optimization speedup and the fitness maximization.

4.1.2. The Speedup Provided by Parallel SGA

Two SGA optimization processes with $p_{BNN} = 0.4$ and 0.6 are considered in this section, and the number of parallel processes in GA fitness calculation is set as 12. Assuming that the cost of running the surrogate model is negligible compared to the CP simulation and the computing time for different CP simulations are similar, the computational time cost ratio of the SGA to the original GA is equivalent to the call count ratio. The number of surrogate (the BNN and the Lookup surrogate model) calls and phenomenological CP simulation calls needed to calculate the fitness in each generation is presented in Figure 10. Consequently, the speedup provided by the SGA method can be estimated with the cumulation of the call counts using the following equation:

$$\frac{1}{n_s} = \frac{t_{SGA}}{t_{GA}} = \frac{\sum_i^n N_{CP}(i)}{\sum_i^n (N_{CP}(i) + N_{surrogate}(i))} \quad (31)$$

where n_s is the speedup achieved using parallel SGA compared to traditional parallel GA optimization, t_{SGA} and t_{GA} are the expected time cost of running the SGA and GA, respectively, $N_{CP}(i)$ and $N_{surrogate}(i)$ are the number of CP simulation and surrogate calls

in generation i , respectively, and n is the current generation. Around the end of the optimization process, when the genes in the population converge to the optimum and become similar to each other, the number of Lookup surrogate calls increases, providing more speedup than could be achieved by only utilizing the BNN surrogate. The speedup n_s of the optimization process under the considered p_{BNN} are 2.917 (Figure 10a) and 3.765 (Figure 10b), respectively, compared to the ordinary parallel GA. When considering the speedup provided by parallel fitness calculation, using the parallel surrogate-assisted GA eventually led to a speedup of around 50 compared to the serial GA on the workstation used in this study.

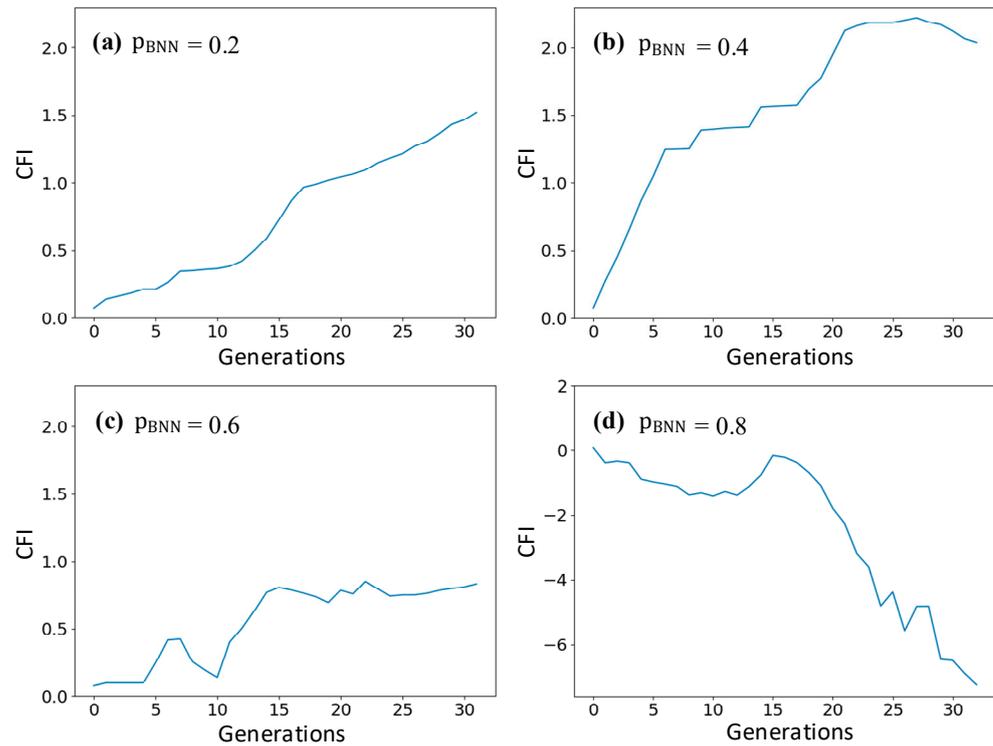


Figure 9. The cumulative fitness improvement in different SGA runs when p_{BNN} equals (a) 0.2, (b) 0.4, (c) 0.6, and (d) 0.8.

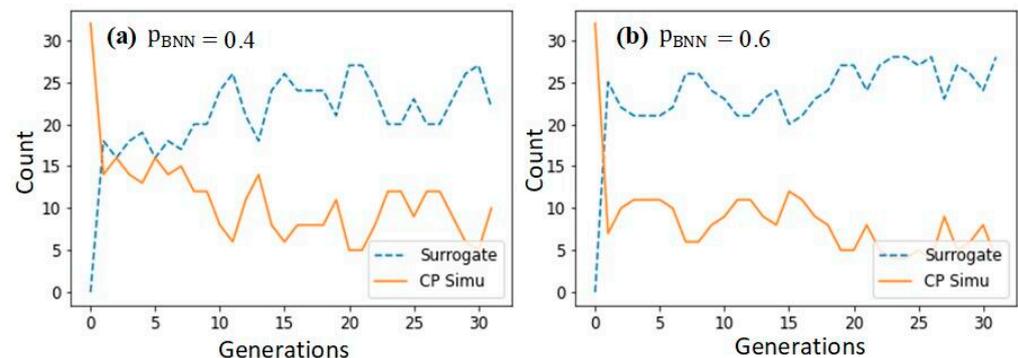


Figure 10. The number of phenomenological CP simulation calls and surrogate (the BNN or Lookup) calls in GA generations (a) $p_{BNN} = 0.4$, (b) $p_{BNN} = 0.6$.

4.2. Calibration Results and Interpretation

In this section, 2 separate SGA searching processes are run for 32 generations using the numerical input mentioned in Section 3 with p_{BNN} being equal to 0.4.

4.2.1. Comparison of Experimental and Simulated Results

The adjustable parameters of the best solution in two runs of the phenomenological and dislocation-density-based CP models are listed in Tables 8 and 9, respectively. The mechanical response and pole figures are compared in Figure 11. For both CP models, the SGA-output best solution in the two runs showed good agreement between CP simulation prediction and the experimentally measured stress–strain curve. The error between the target curve and two simulation runs quantified by the distances defined by Equation (1) is 4.18% and 4.15% for the phenomenological CP model and 2.04% and 1.80% for the dislocation-density-based CP model, respectively. Despite the good fit of the stress–strain curve in optimization, there is a relatively large difference in the optimized constitutive parameter h_0 , a , and especially for n . In run 1, parameter h_0 of the best solution lies in the range [100, 700] MPa, while in run 2, it reaches the upper bound, while for the dislocation-density-based CP model, the parameters show a similar convergence tendency but different converge values in two runs. These two different sets of parameters result in almost the same mechanical response. Parameters D_0 , ρ_{dip} and Q_{cl} show a large distribution range in two runs, while i_s , D_a , τ_0 converge to a single point. The dislocation-density-based CP model involves more parameters to characterize the mechanical response of materials than the phenomenological model, showing fewer constraints of the stress–strain curve data and more uncertainty in the calibration results. This difference in results from different runs is further discussed in Section 4.2.3, with the interpretation of the BNN.

Table 8. The optimal parameters of the phenomenological CP model.

Variable	Units	Value (Run 1)	Value (Run 2)
ζ_0	MPa	141.03	142.68
ζ_∞	MPa	899.36	965.65
h_0	MPa	597.60	700.0
n	-	28.44	8.41
a	-	3.34	4.09

Table 9. The optimal parameters of the dislocation-density-based CP model.

Variable	Units	Value (Run 1)	Value (Run 2)
ρ_0	-	1.910×10^{11}	5.682×10^{10}
$\rho_{\text{dip}0}$	-	0.571	0.391
v_0	$\text{m} \cdot \text{s}^{-1}$	1745.177	3253.913
Q_s	J	3.301×10^{-19}	2.944×10^{-19}
Q_{cl}	J	1.979×10^{-19}	2.577×10^{-19}
τ_0	MPa	53.934	80.728
i_s	-	71.101	56.150
D_0	$\text{m}^2 \cdot \text{s}^{-1}$	2.172×10^{-5}	2.243×10^{-5}
D_a	-	8.595	11.820

Figure 12 shows the comparison of experimental and simulated pole figures using different CP models. Both the pole figures obtained by the phenomenological and dislocation-density-based CP models successfully reproduced the (011) fiber texture observed in the measured one (Figure 12a). The orientation distribution and maximum value of texture intensity agree well with the experimental counterpart.

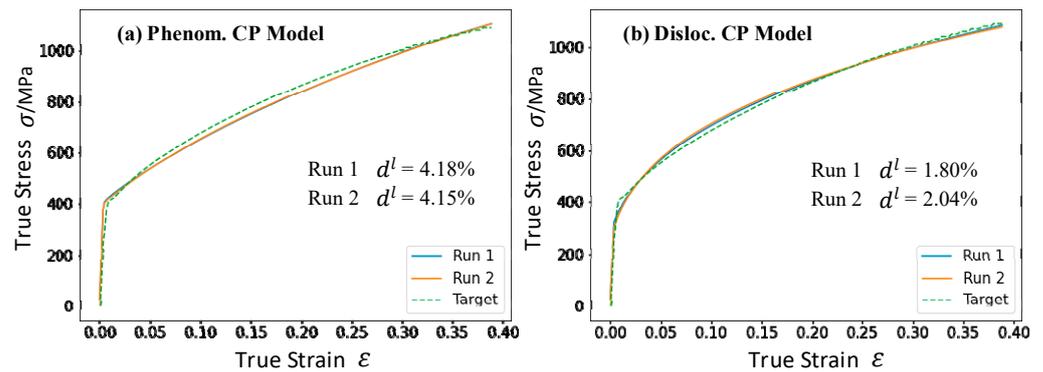


Figure 11. The stress–strain curve of best solution calculated by CP simulation in two runs (run 1, run 2), and the experiment stress–strain curve (target) in (a) a phenomenological CP model and (b) dislocation-density-based CP model. d^l , as defined in Equation (1), denotes the relative difference between simulation-calculated stress–strain curve and experiment stress–strain curve in two runs.

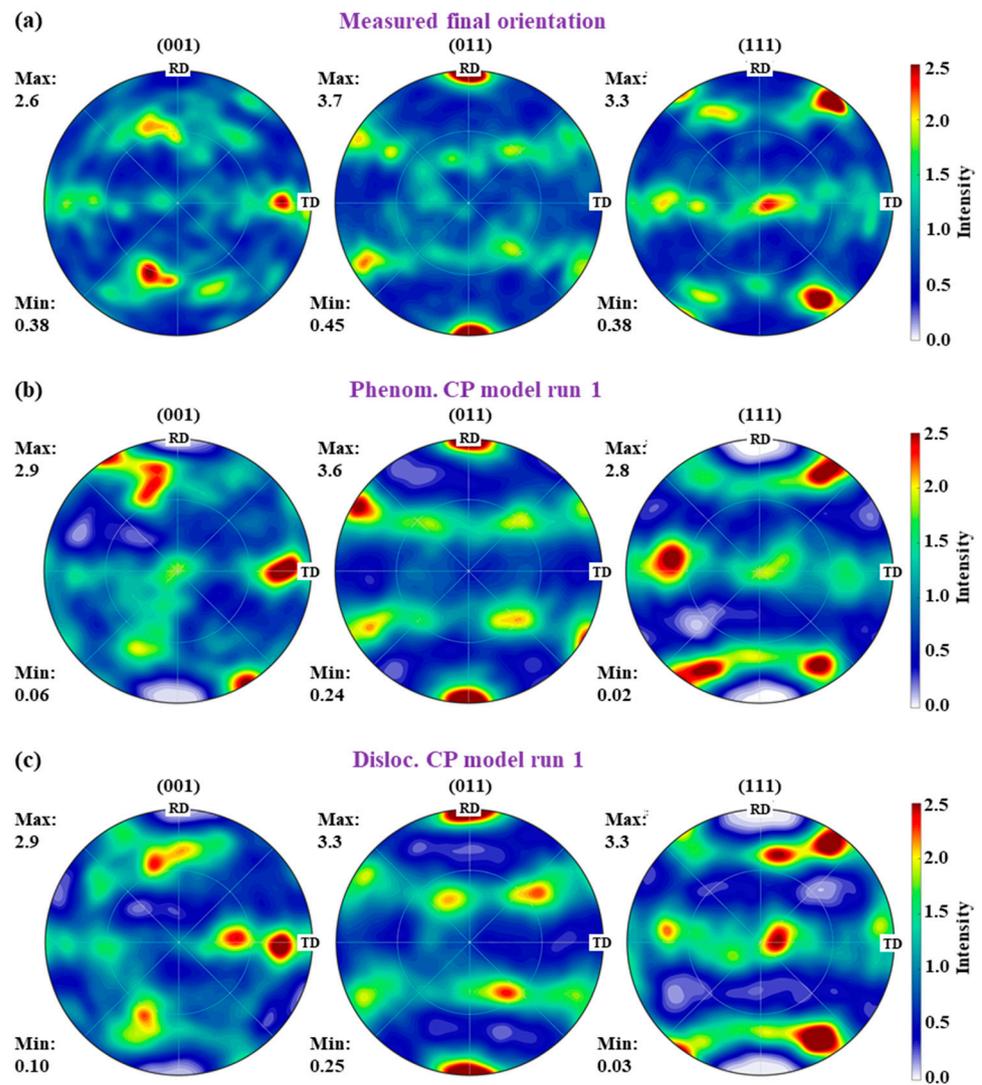


Figure 12. Pole figure of (a) measured final orientation, (b) phenomenological CP model run 1, and (c) dislocation-density-based CP model run 1.

4.2.2. The Gene Evolution

Figure 13 shows the evolution of the gene corresponding to ζ_0 and ζ_∞ in the phenomenological CP model and τ_0 and i_s in the dislocation-density-based CP model. When the searching process stopped at generation 32, the solutions' gene gathered around the same value, suggesting that the global optimum had been found. It should be mentioned that the incremental mutation operator is employed in the GA, which randomly increases or decreases the value of genes in the mutation process. This method is conducive to exploring the parameter space near the optimal solution and simultaneously improving the performance of BNN prediction near the optimum. The fine-tuning of the mutation, with one parameter fixed and another changing, resulted in the cross-like sampling point around the optimum.

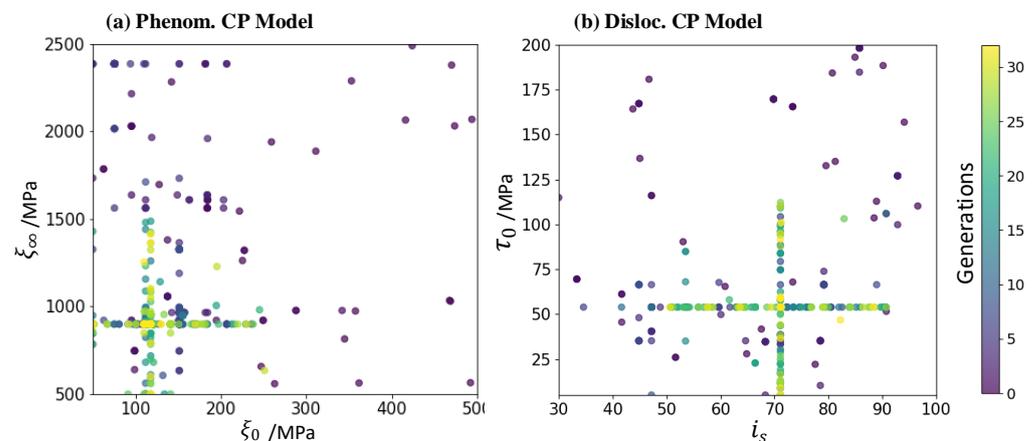


Figure 13. The evolution of the gene corresponding to (a) ζ_0 and ζ_∞ in the phenomenological CP model and (b) τ_0 and i_s in the dislocation-density-based CP model.

This evolution of sample points also indicates the adaptive nature of infill sampling and requires fewer samples to reach the same accuracy around the optimum than the uniform random sampling method in other CP parameters calibration studies. The regions that are not expected to cover the optimum are sparsely sampled, while the interested region is sufficiently probed to make an accurate prediction.

Figure 14 shows the candidate solutions' parameter distribution in the last generation in the two runs of different CP models. It can be observed that the converge value of parameter ζ_0 and ζ_∞ of the phenomenological CP model is consistent in run 1 and run 2, while n shows a wide range of distribution in these solutions and a, h_0 converges to a different point, as shown in Figure 14a,b. For the dislocation-density-based CP model (Figure 14c,d), the optimized parameters in run 1 and run 2 are significantly different. This is caused by two reasons. One is that some parameters have little impact on the mechanical response, and the other is the coupling effects between different CP parameters. This parameter uncertainty is severer in the complex CP model with many parameters to be determined. These effects will be further discussed in Section 4.2.3.

4.2.3. Interpretation by BNN-Integrated Gradients

In this study, the BNN surrogate is not trained to predict the objective directly but first learns to predict stress–strain curve features and then predicts the objective, which provides interpretability for the BNN surrogate model. The integrated gradients [37] are used to analyze the surrogate model, providing insight into the relationship between CP parameters and the stress–strain curve. The integrated gradients can be approximated via

the summation of gradients at small intervals along the path from the baseline x' to the input x :

$$IG_i^{\text{approx}}(x) = \frac{(x_i - x'_i)}{m} \sum_{k=1}^m \frac{\partial F\left(x' + \frac{k}{m}(x - x')\right)}{\partial x_i} \quad (32)$$

where IG_i^{approx} is the approximation of integrated gradients for input parameter x , and m is the number of steps in the Riemman approximation of the integral. In this study, m is set to 50, and the baseline x' is set as 0, which corresponds to the lower bound of the input material parameters. It should be noted that, in this study, the BNN's gradients are sampled in a manner wherein all the parameters increase simultaneously to show the input parameters' global impact on the feature parameters.

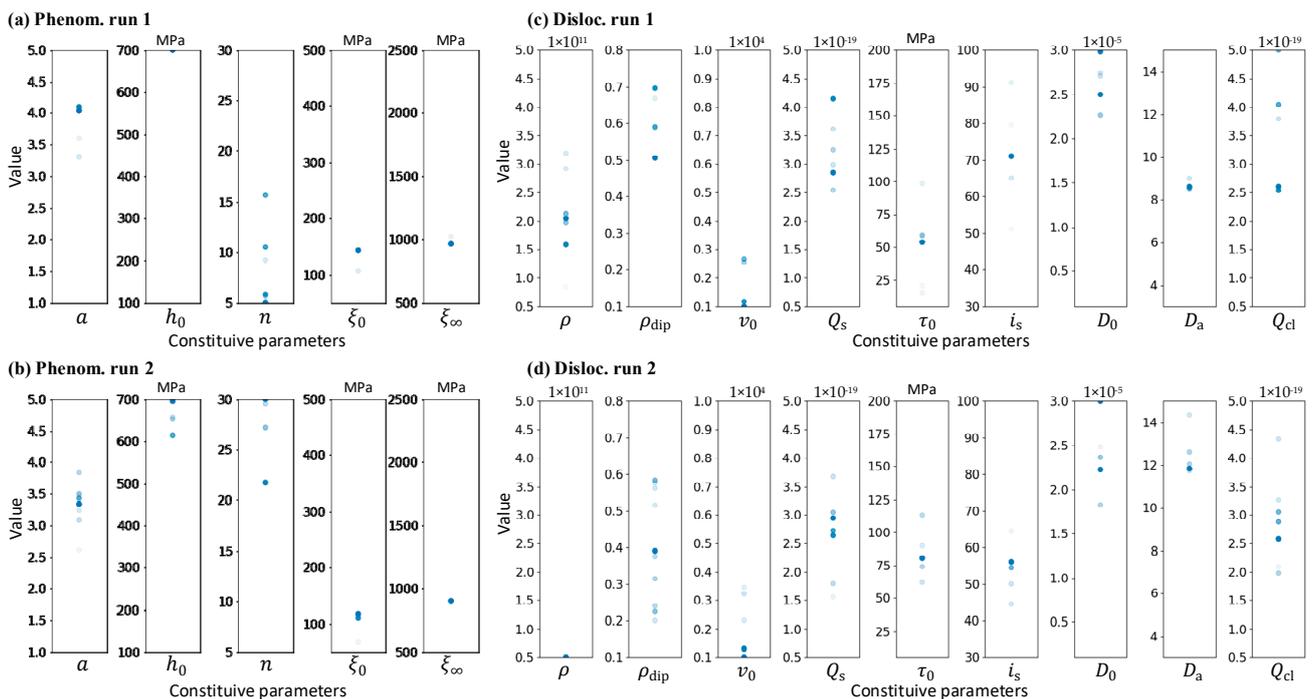


Figure 14. The parameter distribution of the candidate solutions in the last generation in (a) phenomenological model run 1, (b) phenomenological model run 2, (c) dislocation density model run 1, and (d) dislocation density model run 2. The semitransparent blue points represent the certain constitutive parameters of solutions in the last generation, overlap points will show deeper color.

The stress–strain curve shown in Figure 3 is represented as four feature parameters in a normalized form defined in Section 2.2.2 and utilized as the BNN₁ output. Figure 15 shows the integrated gradients of the input parameters of the phenomenological CP model to the output feature parameters. n_p is the most sensitive feature, and controls the curvature of the plasticity part. A larger ζ_∞ or smaller ζ_0 produces a stress–strain curve with a more obvious exponential increase in the plasticity part, corresponding to a larger n_p value. Parameters h_0 and ζ_∞ have a positive contribution to E_1 and a negative contribution to ε_y ; thus, increasing h_0 and ζ_∞ will result in a stress–strain curve with a lower yield point and higher saturation stress. ε_b is indirectly affected by yield and saturate stress, which is shown to be the least sensitive feature in the curve.

In both run 1 and run 2, the constitutive parameter n shows a slight influence on all the feature parameters, and a varying n may not influence the output. This causes difficulty in accurately calibrating parameter n and accordingly shows a wide range of candidate values (Figure 14). This accounts for the large difference in parameter n observed in the results of two runs (Table 8). There also exists a relatively large difference in parameters a and h_0 in the two runs, which is the result of parameter coupling. In Figure 15, the

parameters a and h_0 show an opposite effect on the stress–strain curve features, indicating that these parameters can increase or decrease simultaneously while obtaining almost the same output features. It should be noted that the parameter ε_y , which controls the yield strain, does not contribute much to the fitness calculation; thus, the different effect on ε_y found in run 1 and run 2 is neglectable in parameter coupling. It can be seen that the effect of ζ_∞ is somewhat similar to that of h_0 , indicating a potential coupling between ζ_∞ , h_0 , a . Thus, when h_0 is limited by the upper bound, ζ_∞ may act as its substitution, resulting in a slightly larger value of ζ_∞ (Table 8, run 2). Parameter ζ_0 shows good consistency in two runs, indicating that it was easily determined by the optimization process.

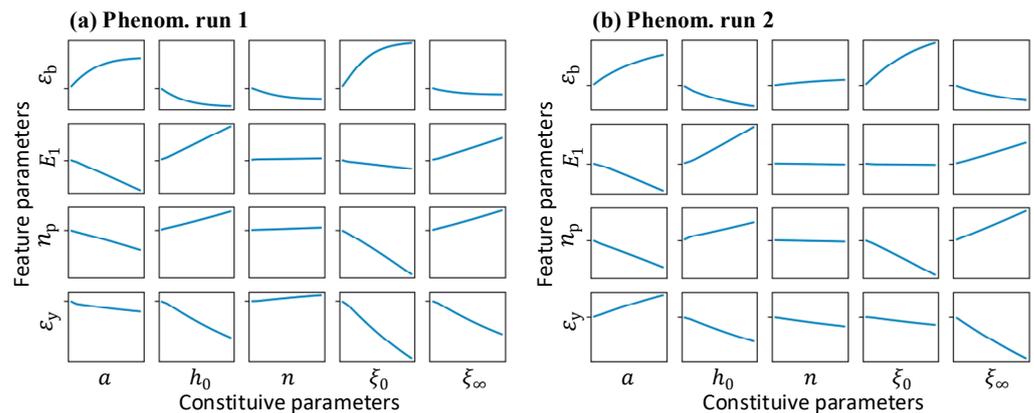


Figure 15. The integrated gradients for the BNN surrogate of the phenomenological CP model in (a) run 1 and (b) run 2.

Figure 16 presents the integrated gradients for the BNN surrogate of the dislocation-density-based CP model. The parameters show more complex coupling between parameters in this calibration compared with the simple two-parameter coupling in the phenomenological CP model. D_a , i_s , and τ_0 are the most influential parameters on the output features; thus, by fitting the stress–strain curve, they show good convergence (Figure 14). However, due to potential coupling with the other parameters, their converge points are influenced. Q_s is a relatively influential parameter and is constrained within a range. ρ_0 , ρ_{dip0} , and Q_{cl} show various effects in different runs. When it shows a greater impact on the feature parameters (characterizing the mechanical response), as seen in Figure 16, its distribution correspondingly has a narrow range, as seen in Figure 14. This results from the different influences of these parameters on different sample regions. The BNN captures this pattern and can successfully explain the distribution of the candidate solution’s parameters.

Though most of the CP parameters cannot be exactly determined by fitting the stress–strain curve and showing coupling or noneffective tendency, they are constrained to a certain extent. The interpretation of the BNN can guide further experiment design to decouple particular parameters and add constraints provided by the attached experiment or prior knowledge. With the knowledge extracted from the BNN, the constraint of the macroscopic mechanical response on the model parameters is outlined, and this can be used to guide future experiment design. In general, a parameter is well constrained by the fitting data and converges to a consistent point when it (a) shows a significant impact on the output features and (b) does not couple with other adjustable parameters. The parameters showing a range distribution in the last generation of the SGA are considered only partly constrained in a range with the experiment data, with a slighter influence on the output features. Though a lack of constraints exists, some parameters can be determined with a macroscopic mechanical response, and others need further constraints (e.g., additional experiment dataset as the fitting target or prior knowledge-based constraints for the GA) or decoupling (e.g., calibrate coupled parameters separately) to be further calibrated.

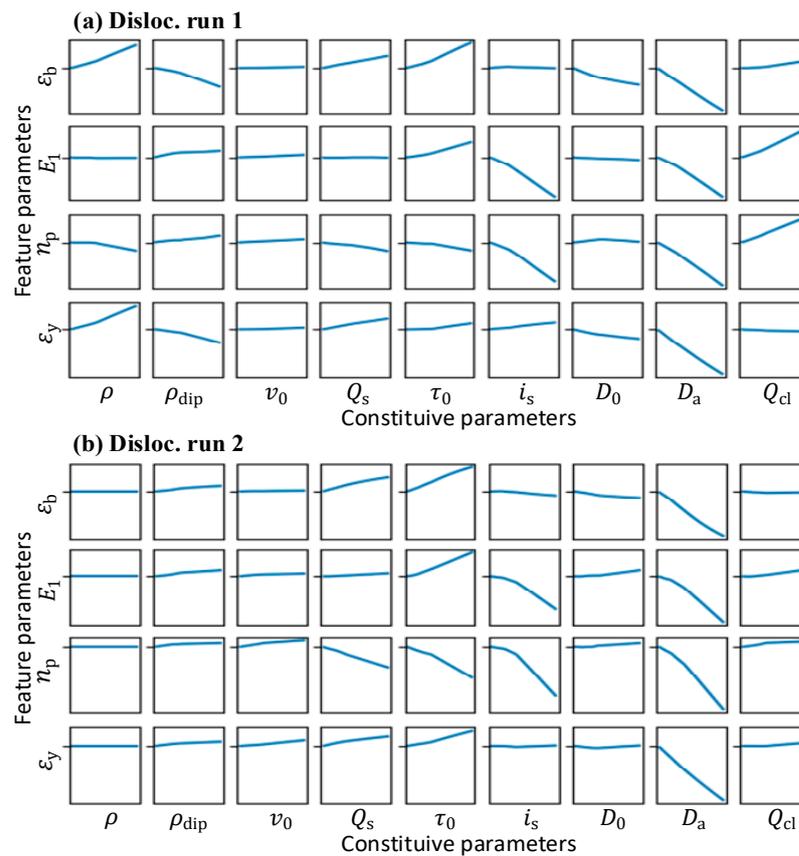


Figure 16. The integrated gradients for the BNN surrogate of dislocation-density-based CP model in (a) run 1 and (b) run 2.

5. Conclusions

We aimed to accelerate the calibration of a CP model and maintain its accuracy simultaneously. A BNN-based surrogate was employed, and the infill sampling (IS) method were utilized to allocate new sample points as well as iteratively update the BNN surrogate during the optimization process. By using the integrated gradients, the evolution of parameters during the calibration process and the coupling effect of different parameters were analyzed. The main conclusions can be drawn as follows:

1. A BNN SGA optimization method was proposed for calibrating crystal plasticity models. The MCDropout-based BNN makes a prediction with uncertainty information, and guides the GA optimization process by infill sampling. The BNN is trained stepwise to predict feature parameters of the mechanical response and provides interpretability for the surrogate.
2. Maximizing EI is used as the infill sampling criteria, determining which candidate solutions should be selected as new sample points and performing the time-consuming simulation. The Lookup-based surrogate, which queries the dataset and directly returns the fitness for similar solutions, is another data-based surrogate utilized in this study.
3. The BNN SGA method is applied to the typical phenomenological and dislocation-density-based CP models of an HEA. The speedup of the optimization process is 2.917 and 3.765 when the proportion of performing BNN surrogates equals 0.4 and 0.6, respectively. Using the identified parameters, the stress–strain curve and pole figure obtained by compression tests were successfully reproduced by CP simulation.
4. This method allows for interpretation of the BNN surrogate using the integrated gradients and extract information of the relationship between input CP parameters and the feature of the mechanical response. The effect of the CP parameters on the

stress–strain curve and the coupling between CP parameters was analyzed. The interpretation of the BNN can guide further experiment design to constrain or decouple particular CP parameters to refine the calibration result.

5. The SGA method proposed in this paper can be extended to calibrate CP models including more complex deformation mechanisms, such as deformation twinning and phase transformation. The BNN surrogate with iterative infill sampling and integrated gradients can also be extended to establish an interpretable and fast multiscale surrogate plasticity model that characterizes the microscale slip, mesoscale strain, and macroscale mechanical response.
6. Like most optimization algorithms, this method relies on an initial parameter range. Due to the complex nonlinear relationship between parameters of the CP model, the range of other parameters will change dramatically if the range of one parameter is modified, so it is not easy to determine the initial range. In this paper, the coupling between CP parameters and the effect of CP parameters on the mechanical response are demonstrated by means of an integral gradient, but no quantitative model is introduced to systematically describe their coupling relationship. There is no quantitative description of how the provided experimental data constrain the results of the calibration. Future research could focus on breaking these limitations and propose a more robust calibration method.

Author Contributions: Methodology, software, investigation, visualization, writing—original draft preparation, S.Y.; conceptualization, funding acquisition, visualization, writing—review and editing, X.T.; data curation, validation, writing—review and editing, L.D., P.G., M.Z. and J.J.; project administration, funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 52090043), the Aeronautical Science Foundation of China (No. 20200036079001), and the National Natural Science Foundation of China (No. 52105337).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tang, X.; Wang, Z.; Deng, L.; Wang, X.; Long, J.; Jiang, X.; Jin, J.; Xia, J. A Review of the Intelligent Optimization and Decision in Plastic Forming. *Materials* **2022**, *15*, 7019. [[CrossRef](#)] [[PubMed](#)]
2. Bertin, M.; Du, C.; Hoefnagels, J.P.M.; Hild, F. Crystal plasticity parameter identification with 3D measurements and Integrated Digital Image Correlation. *Acta Mater.* **2016**, *116*, 321–331. [[CrossRef](#)]
3. Do, B.; Ohsaki, M. Bayesian optimization for inverse identification of cyclic constitutive law of structural steels from cyclic structural tests. *Structures* **2022**, *38*, 1079–1097. [[CrossRef](#)]
4. Herrera-Solaz, V.; LLorca, J.; Dogan, E.; Karaman, I.; Segurado, J. An inverse optimization strategy to determine single crystal mechanical behavior from polycrystal tests: Application to AZ31 Mg alloy. *Int. J. Plast.* **2014**, *57*, 1–15. [[CrossRef](#)]
5. Savage, D.J.; Feng, Z.; Knezevic, M. Identification of crystal plasticity model parameters by multi-objective optimization integrating microstructural evolution and mechanical data. *Comput. Methods Appl. Mech. Eng.* **2021**, *379*, 113747. [[CrossRef](#)]
6. Sedighiani, K.; Diehl, M.; Traka, K.; Roters, F.; Sietsma, J.; Raabe, D. An efficient and robust approach to determine material parameters of crystal plasticity constitutive laws from macro-scale stress-strain curves. *Int. J. Plast.* **2020**, *134*, 102779. [[CrossRef](#)]
7. Saleeb, A.F.; Gendy, A.S.; Wilt, T.E. Parameter-Estimation Algorithms for Characterizing a Class of Isotropic and Anisotropic Viscoplastic Material Models. *Mech. Time-Depend. Mater.* **2002**, *6*, 323–361. [[CrossRef](#)]
8. Andrade-Campos, A.; Thuillier, S.; Pilvin, P.; Teixeira-Dias, F. On the determination of material parameters for internal variable thermoelastic–viscoplastic constitutive models. *Int. J. Plast.* **2007**, *23*, 1349–1379. [[CrossRef](#)]
9. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
10. Furukawa, T.; Sugata, T.; Yoshimura, S.; Hoffman, M. An automated system for simulation and parameter identification of inelastic constitutive models. *Comput. Methods Appl. Mech. Eng.* **2002**, *191*, 2235–2260. [[CrossRef](#)]
11. Pandey, A.; Pokharel, R. Machine learning enabled surrogate crystal plasticity model for spatially resolved 3D orientation evolution under uniaxial tension. *arXiv* **2020**, arXiv:200500951.

12. Pandey, A.; Pokharel, R. Machine learning based surrogate modeling approach for mapping crystal deformation in three dimensions. *Scr. Mater.* **2021**, *193*, 1–5. [[CrossRef](#)]
13. Yang, Z.; Papanikolaou, S.; Reid, A.C.E.; Liao, W.-K.; Choudhary, A.N.; Campbell, C.; Agrawal, A. Learning to Predict Crystal Plasticity at the Nanoscale: Deep Residual Networks and Size Effects in Uniaxial Compression Discrete Dislocation Simulations. *Sci. Rep.* **2020**, *10*, 8262. [[CrossRef](#)]
14. Rezaei Mianroodi, J.; Siboni, N.H.; Raabe, D. Teaching Solid Mechanics to Artificial Intelligence: A fast solver for heterogeneous solids. *arXiv* **2021**, arXiv:210309147.
15. Forrester, A.I.J.; Keane, A.J. Recent advances in surrogate-based optimization. *Prog. Aerosp. Sci.* **2009**, *45*, 50–79. [[CrossRef](#)]
16. Williams, B.A.; Cremaschi, S. Surrogate model selection for design space approximation and surrogatebased optimization. *Comput. Aided Chem. Eng.* **2019**, *47*, 353–358.
17. Ibragimova, O.; Brahme, A.; Muhammad, W.; Lévesque, J.; Inal, K. A new ANN based crystal plasticity model for FCC materials and its application to non-monotonic strain paths. *Int. J. Plast.* **2021**, *144*, 103059. [[CrossRef](#)]
18. Goldberg, D.E.; Holland, J.H. Genetic Algorithms and Machine Learning. *Mach. Learn.* **1988**, *3*, 95–99. [[CrossRef](#)]
19. Gad, A.F. Pygad: An intuitive genetic algorithm python library. *arXiv* **2021**, arXiv:210606158.
20. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *arXiv* **2016**, arXiv:150602142.
21. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer: New York, NY, USA, 1996.
22. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:12070580.
23. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the International Council for Machinery Lubrication, Haifa, Israel, 21–24 June 2010.
24. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
25. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
26. Chen, D.; Li, Y.; Yang, X.; Jiang, W.; Guan, L. Efficient parameters identification of a modified GTN model of ductile fracture using machine learning. *Eng. Fract. Mech.* **2021**, *245*, 107535. [[CrossRef](#)]
27. Parr, J.M.; Keane, A.J.; Forrester, A.I.J.; Holden, C.M.E. Infill sampling criteria for surrogate-based optimization with constraint handling. *Eng. Optim.* **2012**, *44*, 1147–1166. [[CrossRef](#)]
28. Kushner, H.J. A New Method of Locating the Maximum Point of an Arbitrary Multippeak Curve in the Presence of Noise. *J. Basic Eng.* **1964**, *86*, 97–106. [[CrossRef](#)]
29. Liu, J.; Han, Z.; Song, W. Comparison of infill sampling criteria in kriging-based aerodynamic optimization. In Proceedings of the 28th Congress of the International Council of the Aeronautical Sciences, Brisbane, Australia, 23–28 September 2012; pp. 1625–1634.
30. Briffoteaux, G.; Gobert, M.; Ragonnet, R.; Gmys, J.; Mezmaç, M.; Melab, N.; Tuytens, D. Parallel surrogate-assisted optimization: Batched Bayesian Neural Network-assisted GA versus q-EGO. *Swarm Evol. Comput.* **2020**, *57*, 100717. [[CrossRef](#)]
31. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2016**, *104*, 148–175. [[CrossRef](#)]
32. Roters, F.; Diehl, M.; Shanthraj, P.; Eisenlohr, P.; Reuber, C.; Wong, S.L.; Maiti, T.; Ebrahimi, A.; Hochrainer, T.; Fabritius, H.O.; et al. DAMASK—The Düsseldorf Advanced Material Simulation Kit for modeling multi-physics crystal plasticity, thermal, and damage phenomena from the single crystal up to the component scale. *Comput. Mater. Sci.* **2019**, *158*, 420–478. [[CrossRef](#)]
33. Hutchinson, J.W. Bounds and self-consistent estimates for creep of polycrystalline materials. *Proc. R. Soc. Lond. A Math. Phys. Sci.* **1976**, *348*, 101–127.
34. Cheng, J.; Ghosh, S. A crystal plasticity FE model for deformation with twin nucleation in magnesium alloys. *Int. J. Plast.* **2015**, *67*, 148–170. [[CrossRef](#)]
35. Tian, F.; Varga, L.K.R.; Shen, J.; Vitos, L. Calculating elastic constants in high-entropy alloys using the coherent potential approximation: Current issues and errors. *Comput. Mater. Sci.* **2016**, *111*, 350–358. [[CrossRef](#)]
36. Tang, X.F.; Peng, L.F.; Shi, S.Q.; Fu, M.W. Influence of crystal structure on size dependent deformation behavior and strain heterogeneity in micro-scale deformation. *Int. J. Plast.* **2019**, *118*, 147–172. [[CrossRef](#)]
37. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. *arXiv* **2017**, arXiv:170301365.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.