



Deep learning-based phase prediction of high-entropy alloys: Optimization, generation, and explanation

Soo Young Lee^{a,1}, Seokyeong Byeon^{a,1}, Hyoung Seop Kim^{b,c,d}, Hyungyu Jin^{a,*}, Seungchul Lee^{a,e,*}

^a Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

^b Department of Materials Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

^c Center for High Entropy Alloy, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

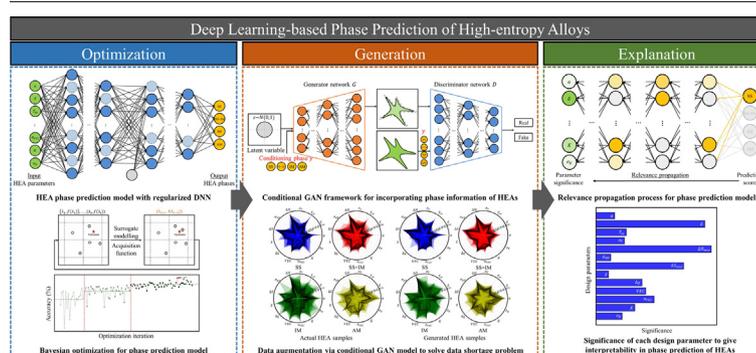
^d Graduate Institute of Ferrous Technology, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

^e Graduate School of Artificial Intelligence, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

HIGHLIGHTS

- We propose deep learning-based methods for reliable phase prediction of high-entropy alloys with three aspects.
- Promising set of hyper-parameters for the regularized deep neural network are searched via Bayesian optimization process.
- Generative design of neural network is established to produce realistic data from pre-existing high-entropy alloy samples.
- Additional samples by generative learning framework can improve phase prediction performance of existing high-entropy alloys.
- Significant design parameters for phase prediction of HEAs are revealed based on the interpretation of our proposed model.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:

Received 7 August 2020

Received in revised form 6 October 2020

Accepted 23 October 2020

Available online 27 October 2020

Keywords:

High-entropy alloys

Phase prediction

Deep learning

Deep neural networks

Bayesian optimization

Conditional generative adversarial networks

ABSTRACT

Identifying phase information of high-entropy alloys (HEAs) can be helpful as it provides useful information such as anticipated mechanical properties. Recently, machine learning methods are attracting interest to predict phases of HEAs, which could reduce the effort for designing new HEAs. As research direction is in its infancy, there is still plenty of room to develop machine learning models to improve the prediction accuracy and further guide the design of HEAs. In this work, we employ deep learning-based methods regarding optimization, generation, and explanation, for enhancing the performance and identifying key design parameters for phase prediction of HEAs. We first establish regularized deep neural networks for predicting HEA phases and optimize hyper-parameters concerning model architecture, training, and regularization. To overcome data shortage of HEAs, we then focus on developing conditional generative adversarial network for generating additional HEA samples. We observe the augmentation from our generative model significantly improves model performance, achieving prediction accuracy of 93.17%. Lastly, we concentrate on understanding contributions of design parameters to identifying solid solution (SS) phase as an example. Our work delivers guidance not only for developing a reliable

* Corresponding authors at: Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea.

E-mail addresses: hgjin@postech.ac.kr (H. Jin), seunglee@postech.ac.kr (S. Lee).

¹ Both authors contributed to this work equally.

deep learning-based phase prediction model, but for explaining significant design parameters to assist design of novel HEAs.

© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

High-entropy alloys (HEAs) refer to multi-element alloys stabilized in a single phase by high configurational entropy of random mixing of elements [1–3]. While traditional alloys typically consist of single or a couple of elements, HEAs are designed to contain multiple constituent elements with equal or similar atomic concentrations. HEAs have received considerable attention for their superior mechanical properties such as hardness, ductility, and corrosion resistivity over those of conventional alloys [4–9]. Very recently, the concept of HEAs has extended into functional materials, such as catalytic materials for hydrogen production, thermoelectric materials, and coating materials [10–12]. The large composition space of HEAs offers a wide range of tunability in terms of physical, chemical, and mechanical properties, making this class of material promising for a variety of application domains. On the other hand, the large composition space also puts a great challenge for designing novel HEAs with tailored properties or functionalities. Because of a large number of possible compositions, the typical trial-and-error approach generally takes considerable time and cost with a low success rate.

When designing a new HEA with desired properties, predicting its phase information is crucial as it is closely linked to particular material properties. The phase of HEAs can be classified into four categories, i.e., solid solution (SS), intermetallic compound (IM), mixed SS and IM (SS + IM), and amorphous (AM) phases. It is known that a SS is desired to obtain high hardness, while an AM phase is needed for better elasticity or corrosion resistivity [3,13,14]. Besides, the IM has been suggested as the preferable phase in terms of the functional materials [15]. Therefore, developing a reliable approach to estimate the phase of HEAs with random compositions is of great importance as the first step towards designing novel HEAs.

The parametric or computational approach has conventionally been used for designing HEAs. In the parametric approach, empirical rules have been established by analyzing the mutual relation between theoretical models and design parameters known so far, e.g., atomic size difference δ , mixing entropy ΔS_{mix} , mixing enthalpy ΔH_{mix} , valence electron concentration (VEC), and electronegativity difference $\Delta\chi$ [16,17]. On the other hand, the computational approaches, e.g., Calculation of Phase Diagrams (CALPHAD), molecular dynamics simulation, and first-principles density functional theory calculations, have been utilized to estimate under what conditions a specific phase is formed and to predict structural and electronic properties of HEAs based on previous experimental data [3]. Those conventional approaches, however, have limitations due to excessive computational costs and high uncertainties [18].

Over recent years, machine learning (ML) has been applied in the field of HEAs as a data-driven approach. The machine learning, which is generally denoted as deep learning (DL) in this paper, possesses feature learning ability from data representation via the complex architecture of neural networks to achieve a global approximation of the nonlinearity. With the development of deep learning algorithms and data accumulation of HEAs, several pieces of research have shown promising results for estimating phases of HEAs. For example, Wen et al. [19] presented an ML-based design strategy to search for new HEAs with high hardness in the Al-Co-Cr-Cu-Fe-Ni system by mapping features of known HEAs and their target properties. Work by Huang et al. [20] suggested several types of ML models for predicting three phases of HEAs, i.e., SS, IM, and SS + IM with five design parameters, i.e., δ , ΔS_{mix} , ΔH_{mix} , VEC, and $\Delta\chi$. By comparing the performance of those models for estimating all the three phases, the authors introduced

a three-layered artificial neural network (ANN) as the most proper model with the best accuracy of 74.3%. While Zhou et al. [21] employed an analogous ANN, the authors considered additional design parameters, e.g., standard deviations of ΔH and VEC among the components, and showed that the existing empirical rules for phase estimation could be appraised by their sensitive analysis of the ANN model.

Although the above studies have shown promising results regarding the DL-guided phase prediction of HEAs, the following challenges remain to fully utilize the approach for practical use, which motivated this study.

- First of all, the optimization of a neural network, which has been suggested as the future work by Huang et al. [20], should be carried out to enhance the predictive performance. Further, various regularization methods should be reflected in the optimization process to determine the most appropriate model without overfitting problems, which are not available in the current literature.
- Secondly, the data shortage problem should be taken into account when using the DL framework, as the number of available HEA samples is limited. While DL can be utilized as a powerful feature descriptor in nonlinear phenomena, its useful application depends heavily on the number of samples [22]. Besides, the problem not only affects the model performance, but also entails poor generalizability considering that there would be a lot more HEAs that have not been discovered yet.
- Lastly, the results obtained by DL algorithms should be explainable to understand the influence of each design parameter in phase estimation. Which parameters largely contribute to the decision of the given DL model is usually considered as a black box. Through a careful interpretation of the neural network, it is possible to explain the effect of each parameter on phase estimation, which could help to design new HEAs.

In order to address the challenges above, this paper suggests a DL approach concerning *optimization*, *generation*, and *explanation*, aiming at enhancing the performance of phase prediction and its explainability. Using gathered data from the previous studies, we first take into account the hyper-parameter optimization of a regularized deep neural network (DNN) to find a promising set of architectural structure and diverse hyper-parameters, e.g., the number of hidden nodes, learning rate, batch size, and dropout rate, while effectively reflecting prior knowledge during the process via the Bayesian optimization (BO) method. In addition, we utilize a conditional generative adversarial network (conditional GAN) to find a model distribution that emulates the distribution of known HEAs, and then to augment realistic samples based on feature representation. We analyze the characteristics of the generated samples and their impact on the performance of the optimized model, showing that the proposed approach yields a significantly improved result for the phase prediction of HEAs. Finally, we evaluate the effect of each design parameter using the propagation-based relevance score of the optimized neural network, in order to elucidate which parameters are influential to the phase prediction of HEAs.

2. Materials and methods

In this section, the dataset we employed for the phase prediction of HEAs is briefly described, followed by overall explanations on DL-based methods, as well as the workflow of the proposed approach.

2.1. Data description

We adopt three different datasets from several previous works [15,20,21,23] for developing our DL-based phase prediction model. After removing redundant samples, we finalize our dataset with a total of 989 samples classified into four phases, i.e., 250 SS, 267 SS + IM, 307 IM, and 165 AM phases. Besides the design parameters hitherto proposed based on the empirical rules, i.e., δ , ΔS_{mix} , ΔH_{mix} , VEC, and $\Delta\chi$ [16,24,25], we capitalize on the additional thermodynamic parameters suggested by Zhou et al. [21], e.g., standard deviation of T_m , ΔH_{mix} , VEC, $\Delta\chi$, and K , where T_m is the average melting temperature and K is the mean bulk modulus. The authors have shown that employing those standard deviations enhances the phase prediction performance by expanding the parameter space [21]. The entire design parameters (denoted as Features) are described in Table 1. Moreover, min-max normalization (ranges from -1 to 1) is conducted to each feature for reflecting every importance on the model equivalently.

To explore the feature distribution of the dataset, we visualize the feature representation of the whole samples in two dimensions using the t-stochastic neighbor embedding (t-SNE) method (see Fig. 1). Herein, a total of 13 design parameters in Table 1 imply 13 dimensions of the feature space, where each sample with those high-dimensional characteristics will be fed into the DNN. Roughly speaking, the t-SNE is operated by embedding high-dimensional data into the low-dimensional space while maintaining the original distributions of the high-dimensional data, i.e., distance similarities among data points [26]. It is worth mentioning that insight into the relative distances among the embedded points can be obtained through the visualization, without giving a physical meaning to each axis. As visualized in Fig. 1, it is shown that the embedded distributions of SS and SS + IM phases are significantly entangled, demonstrating that they have similar feature representations in the original feature space, where it makes it difficult to distinguish between them for the phase prediction. Also, the samples of IM phases are more widely distributed across space, while several points are located near the points of the AM phase as well as the ones of SS and SS + IM phases. Accordingly, our goal is to succeed in nonlinear mapping that differentiates those feature distributions of all the phases via the DL approach.

2.2. Deep neural network and Bayesian optimization

In this work, we mainly construct a DNN, also called a multilayer neural network, to develop a DL model that can classify given data

Table 1

Description of the design parameters used in this study.

Features	Equations
Mean atomic radius	$a = \sum_i c_i \cdot r_i$
Atomic size difference	$\delta = \sqrt{\sum_i c_i \left(1 - \frac{r_i}{\sum_i c_i r_i}\right)^2}$
Average melting temperature	$T_m = \sum_i c_i \cdot T_{mi}$
Standard deviation of melting temperature	$\sigma_T = \sqrt{\sum_i c_i \left(1 - \frac{T_{mi}}{T_m}\right)^2}$
Mixing enthalpy	$\Delta H_{mix} = 4 \sum_{i < j} c_i \cdot c_j \cdot H_{ij}$
Standard deviation of mixing enthalpy	$\sigma_{\Delta H} = \sqrt{\sum_{i < j} c_i \cdot c_j \cdot (H_{ij} - \Delta H_{mix})^2}$
Mixing entropy	$\Delta S_{mix} = -R \sum_i c_i \cdot \log c_i$
Electronegativity	$\chi = \sum_i c_i \cdot \chi_i$
Standard deviation of electronegativity	$\Delta\chi = \sqrt{\sum_i c_i (\chi_i - \chi)^2}$
Valence electron concentration	$VEC = \sum_i c_i \cdot VEC_i$
Standard deviation of VEC	$\sigma_{VEC} = \sqrt{\sum_i c_i (VEC_i - VEC)^2}$
Mean bulk modulus	$K = \sum_i c_i \cdot K_i$
Standard deviation of bulk modulus	$\sigma_K = \sqrt{\sum_i c_i (K_i - K)^2}$

c_i is the stoichiometric ratio of i -th component of alloy, H_{ij} is binary mixing enthalpy in liquid phase, and R is ideal gas constant, respectively.

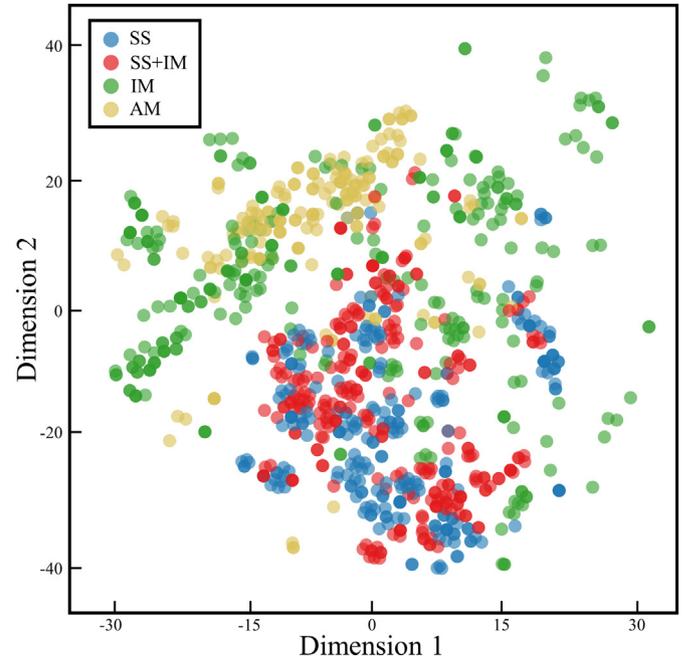


Fig. 1. Feature visualization of the dataset using t-SNE: 13-dimensional features of the entire 989 samples are compressed into a two-dimensional space.

into the four phases of HEAs simultaneously. While our network is designed and trained in a regularized manner for model generalization, we also focus on optimizing its hyper-parameters, e.g., the number of neurons and learning rate, to find the promising DNN architecture as well as learning and regularization settings.

The DNN architecture can be described with consecutive hidden layers, as illustrated in Fig. 2 (a). Whilst located between input (features) and output layer (target phases), these fully-connected layers drive to extract higher-level representations of the features through the training process. Namely, using a series of linear operations between input in the previous layer and trainable parameters, i.e., weights \mathbf{W} and biases \mathbf{b} , the network is trained to produce more useful information for identifying nonlinear phenomena. The operation process can be defined as:

$$x_j^{(l)} = f\left(\sum_i w_{ij}^{(l)} x_i^{(l-1)} + b_j^{(l)}\right) \quad (1)$$

where $x_j^{(l)}$ is the output of j^{th} neuron in the l^{th} layer, $x_i^{(l-1)}$ is the input of i^{th} neuron in the former layer. $w_{ij}^{(l)}$ and $b_j^{(l)}$ denote the weight and bias values assigned for the j^{th} node in the l^{th} layer. f represents the nonlinear activation function at each layer, where a rectified linear unit (ReLU) $f(\cdot) = \max(0, \cdot)$ is utilized [27]. In addition, we use a cross-entropy (CE) loss function, which compares the target and predicted phases in the output layer, while the weights and biases are updated based on backpropagation and gradient descent algorithm in a way that the value of the loss function is minimized.

Particularly, we employ several regularization methods for our DNN model. The regularization is a strategy that fine-tunes the balanced complexity of the neural network between trained and untrained cases. Since the generalization of the model is necessary for better performance, we emphasize on considering the regularization methods to obtain the prediction model that can be expanded to the unseen observations. Firstly, we consider L2 regularization in terms of the loss function, which is designed to impose a penalty on the weights. Since the overfit of the neural network frequently occurs when the weights increase, penalizing the weights as follows could help to suppress the overfitting issue.

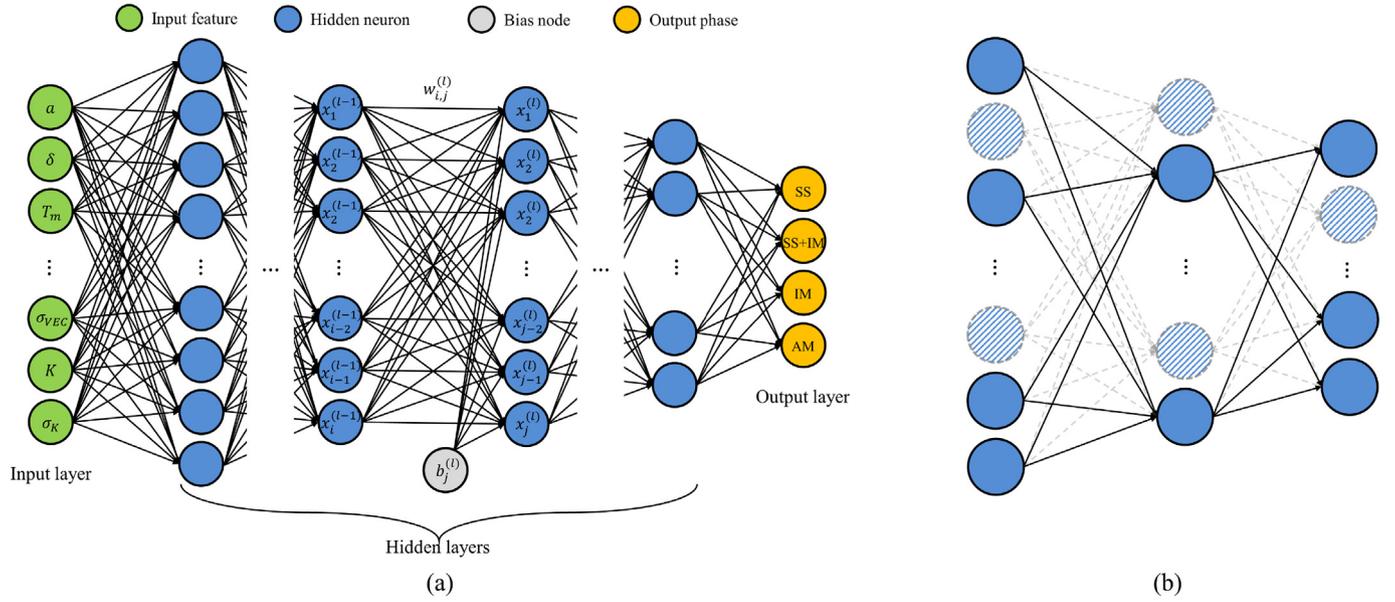


Fig. 2. Schematic diagram of the deep neural network: (a) an architecture of DNN model comprised of input, hidden, and output layers (b) dropout regularization method that controls the connection of the neurons in the hidden layers.

$$L_{penalized} = L_{CE} + \frac{\alpha}{2} \sum_{l=1}^L \|\mathbf{W}^{(l)}\|^2 \quad (2)$$

where $\|\mathbf{W}^{(l)}\|$ is the Frobenius norm of the weight matrix \mathbf{W} in the l^{th} layer. Weight decay coefficient α controls the intensity of the regularization, wherein the more substantial the value is, the more penalties are imposed. Secondly, the dropout technique is utilized by controlling the connection mechanism of the hidden layers within the network [28] (See Fig. 2 (b)). During the training session, it works by turning off random neurons based on the predetermined probability p . The model capacity can be limited as the part of neurons used as feature descriptors are omitted. It is also known to bring about the unbiased fitting of the network, engaging irregular combinations of the selected neurons at each iteration. Likewise, we incorporate these regularization methods into our DNN model for improving the generalization capability that is directly related to the model performance.

Based on the methods above, tuning the hyper-parameters that affect the model structure, training process, and regularization is crucial to building high-performing DNN. It is worth mentioning that searching the hyper-parameters should be mutually considered as the hyper-parameters have complex interactions among each other for the model performance. For instance, an ensemble of L2 regularization and dropout should be controlled appropriately to achieve a well-generalized network [29], while both methods regularize the model in different ways. Also, the number of hidden layers and hidden neurons not merely determines the model capacity but is linked to the regularization techniques concerning underfitting and overfitting problems [30]. While the gradient descent update with L2 regularization, $\mathbf{W} \leftarrow \mathbf{W} - \eta \left(\frac{\partial L}{\partial \mathbf{W}} + \alpha \mathbf{W} \right)$, correlates the two factors, i.e., learning rate η and weight decay coefficient α , it is also known that the learning rate and batch size are related to the generalization ability in mini-batch gradient descent algorithm [31]. Therefore, we propose to consider several hyper-parameters for the phase prediction model of HEAs, as shown in Table 2. We use quantized uniform distribution as the searching space of each hyper-parameter, except the learning rate and weight decay coefficient (log-uniform). Note that the upper/lower bounds of the hidden layer are halved in our experiment if the hidden layer index increases,

Table 2

Hyper-parameters regarding the model structure, training settings, and regularization techniques and their searching spaces for the BO approach. Note that quantized uniform distribution is used to sample the hyper-parameters, except that log-uniform distribution is applied to sample learning rate and weight decay.

Optimization type	Hyper-parameters		Searching space
Model structure	The number of hidden layers	n_l	[1, 6]
	The number of hidden neurons	$n_h^{(l)}$	[128, 512]*
Training settings	Learning rate	η	[1e-5, 1e-1]
	Batch size	N	[2, 30]
Regularization	Weight decay coefficient	α	[1e-6, 1e-3]
	Dropout rate	p	[0.1, 0.7]

* The interval at the first hidden layer.

e.g., [64, 256] for the second hidden layer. Besides, it is worth mentioning that we empirically notice the batch size larger than 30 can deteriorate the learning stability and overall performance of the DNN model in this task.

The Bayesian optimization (BO) approach is utilized to explore hyper-parameters of the DNN. The BO is one of the hyper-parameter optimization methods as a way of automated machine learning, which aims at designing ML models with ML methods [32]. The existing systematic approaches regarding the hyper-parameter optimization are grid search and random search, where the former is to set a uniform interval within the searching space while the latter randomly extracts the candidates from the space (see Fig. 3 (a)). However, the main drawbacks of such methods are that the computational cost grows exponentially as the number of the hyper-parameters increases and that the searching process relies on the randomness [33]. For this reason, we take advantage of the BO as a principled approach that can reflect prior knowledge during the process.

As illustrated in Fig. 3 (b), the BO process is performed based on the surrogate model of the observed samples and evaluations $(\lambda_1, f(\lambda_1)), \dots, (\lambda_n, f(\lambda_n))$ as well as the acquisition function to find a new set of hyper-parameter λ_{t+1} that is likely to be useful for the next exploration. More specifically, the BO problem can be expressed as:

$$\lambda^* = \arg \max_{\lambda \in \Lambda} f(\lambda) \quad (3)$$

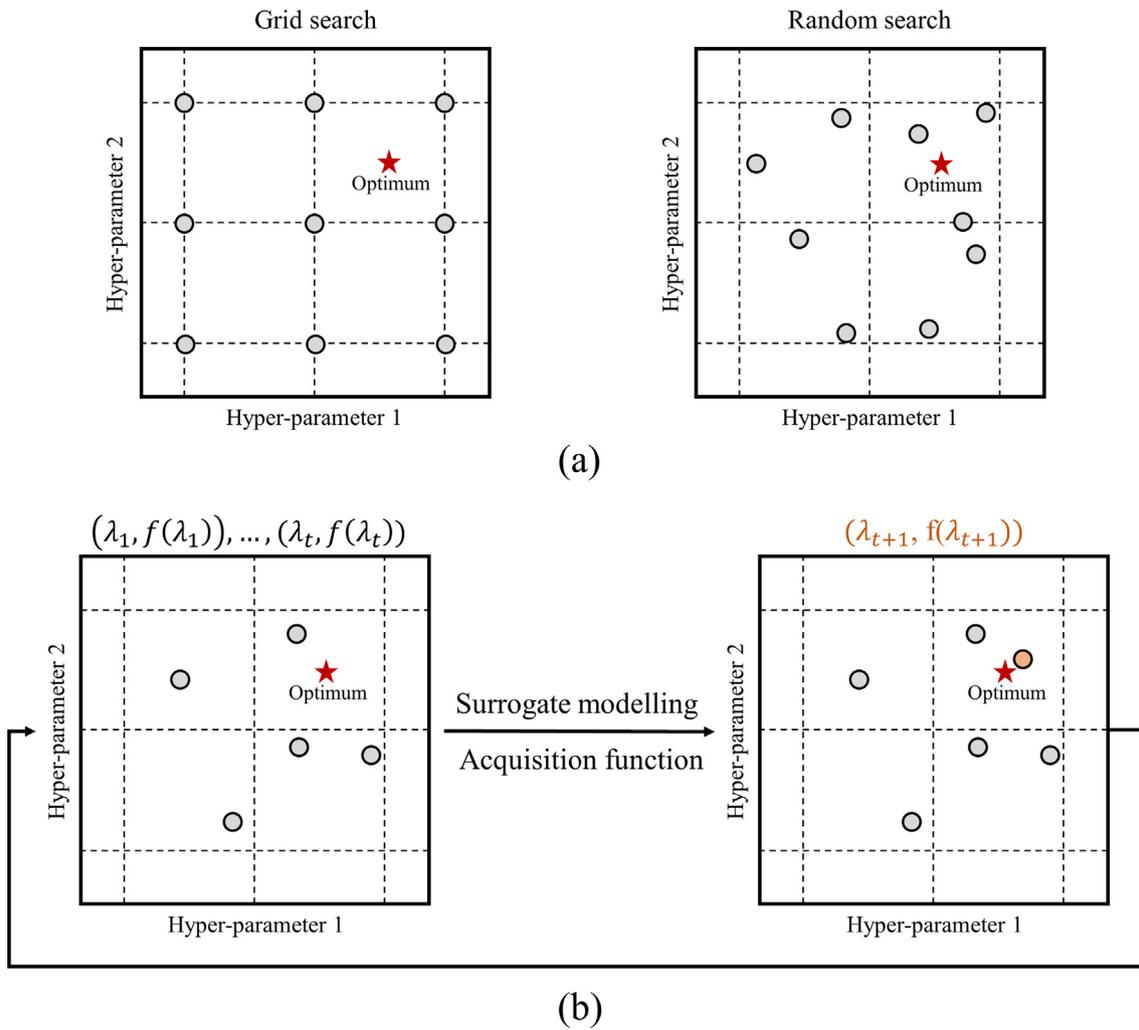


Fig. 3. Schematic descriptions of hyper-parameter optimization methods with two hyper-parameters: (a) comparison between grid search and random search, (b) Bayesian optimization process using surrogate modeling and acquisition function.

where Λ is an entire searching space of the hyper-parameters, while λ is sampled from the bounded regions Λ . $f(\lambda)$ is the evaluation result corresponding to the observed set of the hyper-parameter λ , where the averaged testing accuracy of 4-fold cross-validation is used for our experiment. To measure the generalization ability of each explored model for the result $f(\lambda)$, we employ an early-stopping strategy that ceases the learning promptly when the validation loss starts to increase. Then, the surrogate model probabilistically estimates the unknown objective function f based on the input and output gathered so far, where we develop our surrogate model using tree Parzen estimator (TPE) [34]. Subsequently, the acquisition function plays a role in recommending the next set of hyper-parameter λ_{t+1} that is worthy of exploring or could be useful to find the optimal value, based on the exploration and exploitation strategies from the surrogate model. We use the expected improvement (EI) as the acquisition function [35]. In the end, through the iterative BO process, we finally adopt the set of hyper-parameter λ^* that yields the highest testing accuracy for the phase prediction of the HEAs.

2.3. Conditional generative adversarial network

Using generative adversarial network (GAN), generative modeling of the neural network is established to solve the data shortage problem of HEAs. Unlike the deterministic approach of deep learning, e.g.,

classification, the GAN [36] is designed and trained to find a distribution that approximates the distribution of real data for a generation. The technique has recently been introduced to the materials science field and utilized for materials design and discovery [37–41]. The GAN mainly has a distinction in that it has two different sub-networks, i.e., discriminator D and generator G , and those two opposing networks are trained and optimized in an adversarial learning framework as follows.

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (4)$$

Discriminator D is the network that can distinguish input data as genuine or fake, where it gives a high probability to actual data. On the other hand, generator G is constructed to create the same dimension of data as real input by decoding a random latent variable z . It is trained to minimize the likelihood of the discriminator to classify correctly, where it aims at forming model distribution $p_{model}(x)$ that is analogous to the data distribution $p_{data}(x)$. Through the min-max optimization of the sub-networks, the discriminator becomes better at differentiating real from generated data, whereas the generator learns to produce realistic data to deceive the discriminator. As a result, additional samples considering the distribution of the data are generated from the approximated model distribution $p_{model}(x)$.

Based on the GAN framework, we utilize the conditional GAN to combine the phase information of HEAs into the model (see Fig. 4).

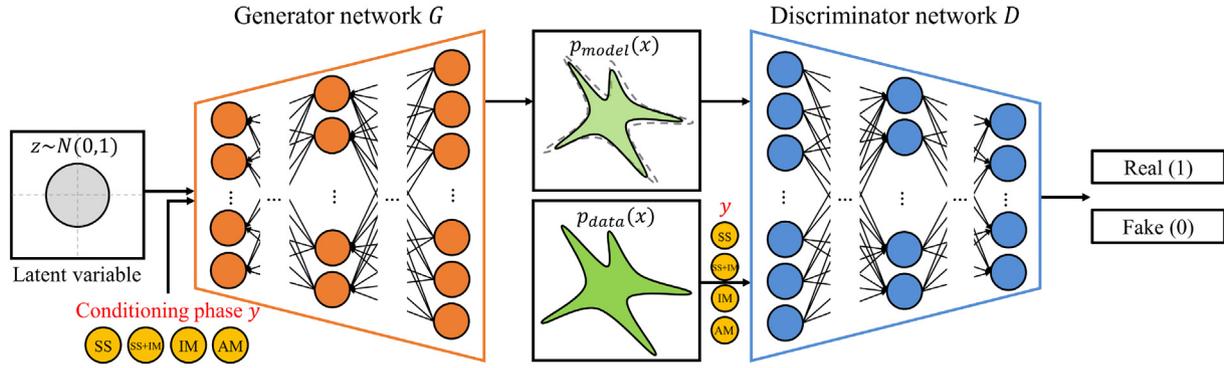


Fig. 4. An architecture of conditional GAN comprised of generator and discriminator sub-networks. Phase information on class labels of SS, SS + IM, IM, and AM phases are incorporated into the generative network to obtain the desired modes from the trained model distribution.

The conditional GAN is the modified version of the GAN, where it takes advantage of additional information, e.g., class label, for controlling the modes of the generated data [42]. The objective function of the conditional GAN can be defined as:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x|y)] + E_{z \sim p_z(z)} [\log (1 - D(G(z|y)))] \quad (5)$$

where inputs of G include conditioning phase labels y and latent variables z randomly extracted from normal distribution $N(0, 1)$, while the inputs of D are either actual HEA data x with y or generated HEA data from G , i.e., $G(z|y)$. The output of the conditional GAN is a binary vector that describes two different types of data, i.e., $D(x|y)$ and $D(G(z|y))$ as real (1) or fake (0). As for the generator network, the combined representation $G(z|y)$ encourages the model to generate samples from the latent space in consideration of the conditioning information y . In our task, since all the phases contain different proportions of data as well as varying feature distributions, we therefore suggest that the phase information, i.e., class labels of SS, SS + IM, IM, and AM, should be conditioned for avoiding the mode-collapse problem and acquiring the feasible results.

The proposed architecture of the conditional GAN is summarized in Table 3. As previously mentioned, the input layer of the two sub-networks reflects the phase of the HEAs as the conditioning variable. This is followed by four dense hidden layers in both generator and discriminator networks, where higher capacity with more hidden neurons

than the discriminator is assigned to the generator network. It should be noted that the generator network is constructed with an appropriate over-all capacity, where a proper amount of dimension is imposed on the latent space for successful generation of the realistic HEA data in this work. Also, we utilize batch normalization [43,44], activation function, and dropout, which are appended between every hidden layer. ReLU and LeakyReLU [45] are employed as the main activation functions of the generator and the discriminator, respectively. Besides, we make use of the dropout method to improve the stability of the training and to generate unbiased samples, where 0.4 for each dropout rate was adopted. Particularly noteworthy is that label smoothing [46] is carried out in the training process to prevent the discriminator from being overconfident on the correct answer. We replace the label of the real sample with 0.9 as a way of label smoothing. Lastly, we jointly train generator and discriminator networks in an adversarial manner by updating each sub-network's trainable parameters alternately, that is, updating the generator network first and then updating the discriminator network, for each iteration. During the training process, we use AdamOptimizer [47] with 0.0002 and 0.0001 learning rates of generator and discriminator, respectively.

3. Results and discussion

In this section, experimental results and discussion for the phase prediction of HEAs are given from perspectives of optimization,

Table 3

The proposed conditional GAN structure in our experiment. The phase type of HEAs is conditioned in both generator and discriminator networks, while each sub-network is established with four hidden layers with different capacity of the neural network.

Generator network			Discriminator network		
Layer	Type	Dimension	Layer	Type	Dimension
Input	Latent + Phase	40 + 4	Input	Feature + Phase	13 + 4
Hidden 1	Dense layer	64	Hidden 1	Dense layer	64
	Batch normalization	64		Batch normalization	64
	ReLU activation	-		LeakyReLU activation	-
	Dropout	-		Dropout	-
Hidden 2	Dense layer	207	Hidden 2	Dense layer	100
	Batch normalization	207		Batch normalization	100
	ReLU activation	-		LeakyReLU activation	-
	Dropout	-		Dropout	-
Hidden 3	Dense layer	128	Hidden 3	Dense layer	50
	Batch normalization	128		Batch normalization	50
	ReLU activation	-		LeakyReLU activation	-
	Dropout	-		Dropout	-
Hidden 4	Dense layer	36	Hidden 4	Dense layer	14
	Batch normalization	36		Batch normalization	14
	ReLU activation	-		LeakyReLU activation	-
	Dropout	-		Dropout	-
Output	Dense layer	13	Output	Dense layer	1
	Tanh activation	-		Sigmoid activation	-

generation, and explanation. We first present the optimization of DL-based phase prediction model, which has been emphasized as necessary in the previous study [20]. By exploring the hyper-parameters for regularization methods as well as the model structure and learning setting via the BO process, we search the regularized DNN model to improve the generalization of the model. Secondly, we employ the conditional GAN to generate realistic HEA data and to augment the number of entire HEA dataset. The main goal of the conditional GAN is to address the data shortage problem of HEAs, by producing additional HEA data that can be used to improve the performance of the DNN-based phase prediction model. Lastly, we explain the impact of each design parameter for the phase prediction of HEAs by applying the propagation-based relevance method to our optimized and enhanced DNN model.

3.1. Experimental setup

We structure our experiments with two strategies for DL-based phase prediction and generation, as schematically shown in Fig. 5. The experiments are mainly classified into two approaches, i.e., optimization of DNN model for phase prediction of HEAs and development of conditional GAN model for data generation, where they are independently established with different networks. As regards the phase prediction model, the DNN model is trained and validated using stratified 4-fold cross-validation. While we randomly sample five initial hyper-parameter sets for the surrogate modeling from our defined searching spaces in Table 2, we conduct a total of 100 iterations of the Bayesian optimization process for obtaining the meaningful hyper-parameters concerning network architecture, training, and regularization settings

from their searching spaces (see Table 2). After the optimization process, the set of hyper-parameters that yields the highest performance is adopted as our proposed DNN model.

On the other hand, the conditional GAN is established via a random search for the data generation model. We consider the entire 989 samples and their conditioning phases y for the training process of the conditional GAN model, followed by the model evaluation. It should be noted that the searching dimensions of the conditional GAN model are considered as follows: [4, 64] for the latent dimension and [8, 256] for each hidden layer's dimension in generator and discriminator network, respectively. In terms of the evaluation procedure, we assume that the conditional GAN is reasonably optimized when the generated 200 HEA data for each phase, i.e., SS, SS + IM, IM, and AM, are fed into the proposed phase prediction DNN model and then show similar testing accuracy to that model that is trained with only actual HEA data. We allow 3% of testing accuracy as an error. In this way, we found the proposed generation model, previously introduced in Table 3. Lastly, we not only analyze the data generated from the conditional GAN model but also incorporate those produced samples into the DNN model to improve the performance and generalization for the phase prediction of HEAs.

Note that we use NVIDIA GeForce RTX 2080 Ti for training both DNN and conditional GAN models, and herein *PyTorch* library is utilized.

3.2. Optimization of DNN-based phase prediction model

Based on the searching spaces of the hyper-parameters as defined in Table 2, the BO process is carried out to find the most appropriate DNN model for predicting all the four phases of HEAs. In order to verify the

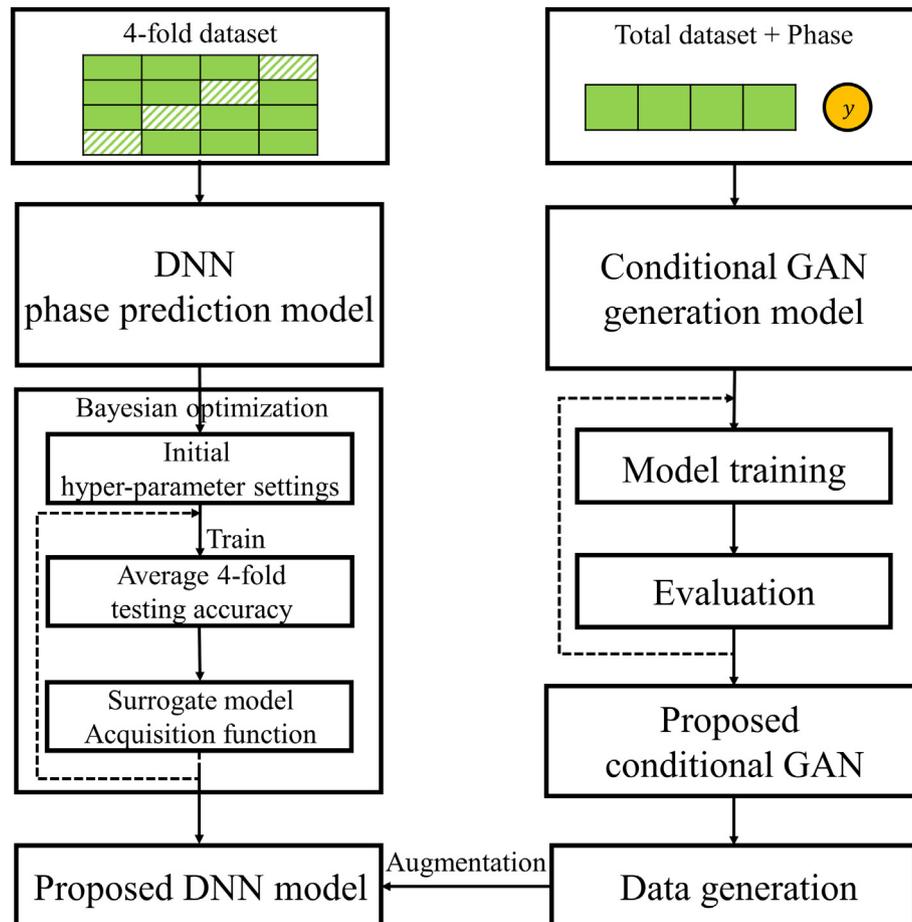


Fig. 5. A schematic flow of the experiments for DL-based phase prediction and data generation model of HEAs: two approaches include the Bayesian optimization process for optimizing DNN-based phase prediction models as well as the conditional GAN approach for generating and augmenting realistic HEA data.

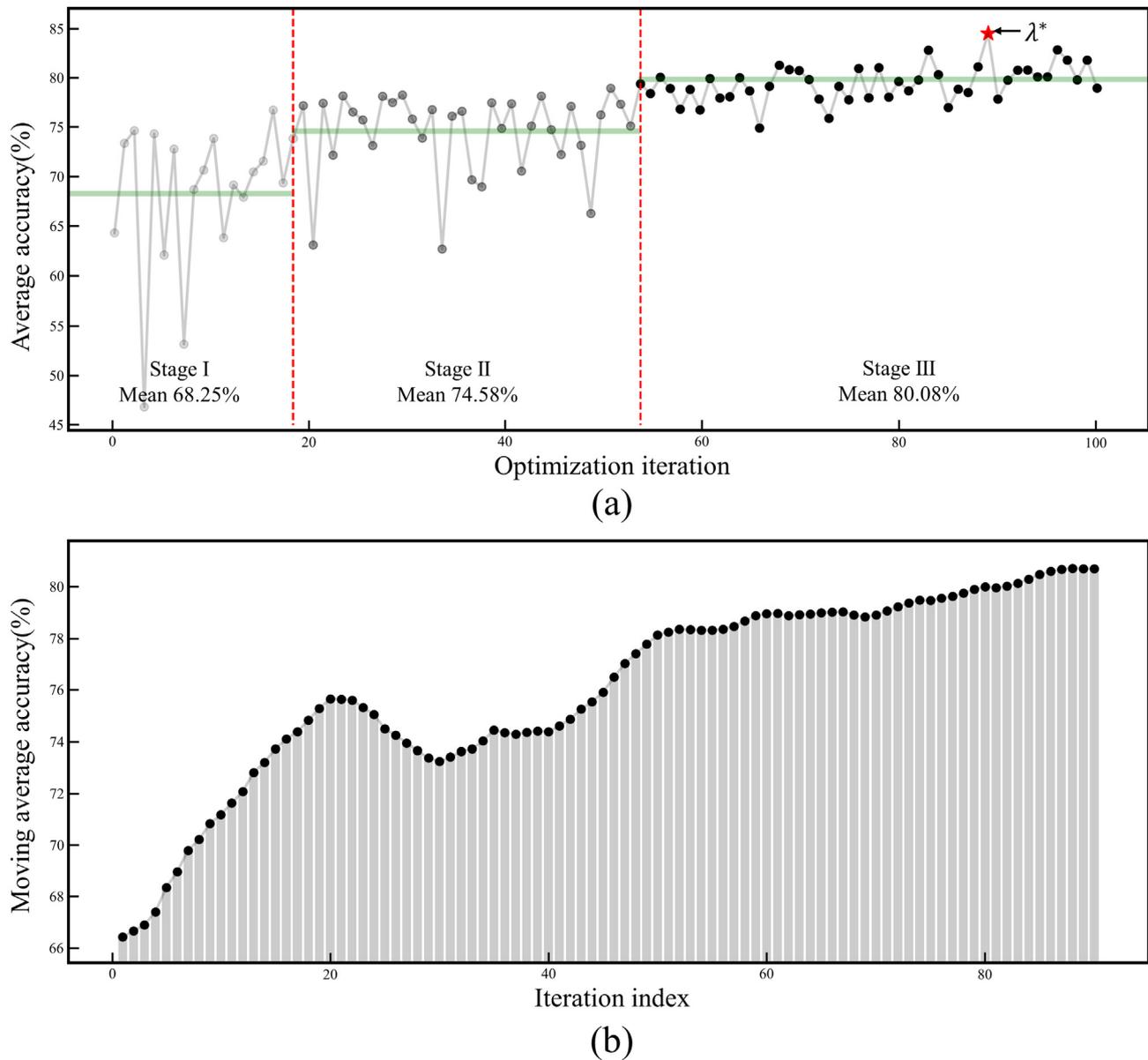


Fig. 6. Progress plots showing the performance trend over 100 iterations during the BO process: (a) average accuracy versus iteration plot. Dashed vertical lines are arbitrarily plotted to visualize the improvement of the average accuracy over BO process, while the green horizontal line represents a mean value of the average accuracy in a certain divided stage. (b) moving average accuracy versus segment index plot, where the average of the accuracy within the last ten points of iterations is considered. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

BO method for searching the hyper-parameters, Fig. 6 (a) presents the averaged 4-fold cross-validation accuracy of the hyper-parameter sets as the optimization progresses. Since the BO method benefits from the useful reflection of the prior knowledge as previously mentioned, it is observed that the performance of the hyper-parameter settings being explored gradually increases over the iteration. As shown in Fig. 6 (a), we arbitrarily plot vertical dashed lines to divide the sections for visualizing and understanding the improvement of the BO process, while the green horizontal line stands for the mean value of evaluations in each divided section (denoted as Stage). It is observed that the second interval of the iteration (Stage II) with a mean accuracy of 74.58%, shows a 6.33% increase from the first one with 68.25%. Besides, it is shown that the average accuracy of Stage III reaches 80.08%, showing considerable improvement (11.83%) over the primary stage. Furthermore, it is also found that the results in Stage I and II are relatively variant, whereas less variant performances are obtained in Stage III. This indicates that the BO process we used focuses more on the suitable hyper-parameter

sets by successfully utilizing the prior knowledge from the surrogate model and acquisition function. We adopt as our proposed set λ^* the hyper-parameter set that achieves the highest testing accuracy of

Table 4

The proposed hyper-parameter set λ^* for model structure, training settings, and regularization, where it is explored by the BO process.

Proposed hyper-parameter set λ^*			
Model structure	The number of hidden layers	n_l	5
	The number of hidden neurons	$n_h^{(1)}$	150
		$n_h^{(2)}$	128
		$n_h^{(3)}$	80
		$n_h^{(4)}$	30
Training settings	Learning rate	η	0.00014
	Batch size	N	15
	Regularization	Weight decay coefficient	α
Dropout rate		p	0.3

84.75% among the iterations. Besides, a moving average plot of the accuracy (Fig. 6 (b)) effectively illustrates the progressive performance improvement of the BO process.

Table 4 describes the proposed hyper-parameter set λ^* of model structure, training settings, and regularization that shows the most promising result for the phase prediction of the HEAs from the BO process. We find that our proposed DNN is structured with a total of five hidden layers, where the number of hidden neurons $n_h^{(l)}$ reduces progressively as the network gets deeper. As for the learning rate η and batch size N , which are both related to the training settings, we obtain 0.000503 and 15, respectively. Also, the ensemble regularization settings with 0.000867 weight decay coefficient α and 0.3 dropout rate p are investigated via the optimization process.

Subsequently, we compare the proposed DNN model's performance with that of several comparative models that utilize simpler machine learning-based classification algorithms, as described in Table 5. As for the comparative models, we consider three different kinds of classifiers, i.e., support vector machine (SVM), decision tree, and XGBoost. Each comparative machine learning-based model is trained and evaluated using *Sklearn* library, where the BO process with 100 iterations is also adopted to tune the hyper-parameters respectively. Details on these baseline models' searching spaces and the explored hyper-parameter sets are presented in Supplementary Table 1. As shown in Table 5, it is observed that the machine learning-based simpler classifiers yield low prediction accuracy, showing at most 72.06% of average test accuracy. On the other hand, we observe that our proposed DNN model that takes advantage of the deep learning-based method and the BO process to find the most promising hyper-parameter set outperforms the comparative machine learning models. This confirms the proposed method's suitability for enhancing the phase prediction accuracy of HEAs compared to the machine learning-based existing approaches.

In addition, it should be noted that our proposed DNN achieves the generalization of the model using regularization methods while maintaining its large complexity with five hidden layers. In order to determine the influence of the regularization methods, we compare the proposed model in Table 4 (denoted as Regularized) and a model with the same structure and training settings but without employing the regularization methods (denoted as Baseline). Fig. 7 visualizes the loss and accuracy curves of the validation set in both models during the training process. The proposed model using the regularization techniques shows a steady decline in the validation loss, while it attains approximately 85% validation accuracy. On the other hand, it is shown that the baseline model where the regularization techniques are omitted induces the overfitting problem as the validation loss rapidly increases. Although the validation accuracy of the baseline model increases as well as the proposed model, it is worth mentioning that not only the maximum performance reaches less than that of the proposed model, but the confidence in the decision-making of the baseline model decreases given the rising behavior of the validation loss. Therefore, we suggest that the performance and generalization ability for estimating the phase of HEAs could be enhanced by incorporating the regularization methods into the prediction model.

To further analyze the phase prediction of HEAs with the DNN, we examine the feature representation of the samples at each layer in our

proposed DNN model. Fig. 8 sequentially represents the embedded feature distributions of the entire samples that are extracted from the input layer and five hidden layers using the t-SNE method. As shown in Fig. 8, the distributions of the HEA phase are entangled in the input layer (same as Fig. 1). While the features of four phases are hardly distinguished in the first hidden layer, interestingly, the distributions of IM and AM phases firstly begin to be divisible by our DNN model from the second hidden layer. Afterward, our model produces more condensed feature representations of IM and AM phases from the hidden layer 3 to 5. From the previous semi-empirical rules [48,49], it is known that IM and AM phases are distinguished from SS and SS + IM phases in terms of atomic size difference. Besides, AM phase has more distinctive features compared to SS and IM, such as lower mixing entropy and enthalpy [3]. These physically distinctive features of IM and AM make them separable from the cluster of four phases in the early hidden layers. As for the SS and SS + IM phases, even though the feature distributions of SS and SS + IM are still entangled in the hidden layer 3, the distributions of them become more distinguishable from the hidden layer 4 to the last hidden layer. It is also known that the SS and SS + IM phases contain physically similar characteristics, e.g., thermodynamic features, considering that the SS + IM phase is a mixture of SS and IM phases at the same time [50], hence make their embedded representations to be more entangled. Although we observe slightly overlapping representations of SS, SS + IM, and IM in the hidden layer 5, the feature representations of four phases are feasibly divided in the last hidden layer. We conclude that the feature distributions of all the phases become separable as they pass through the hidden layers of our trained DNN model, allowing better estimation of phase.

3.3. Generation and augmentation via generative model

We now focus on generating additional samples to augment the number of data for the phase prediction of HEAs. Based on the proposed generative model in Table 3, the generator network is trained to yield an approximated model distribution by considering a 13-dimensional distribution of the existing data and conditioning the phase information. Fig. 9 graphically displays the distributed characteristics of the 13 features for the actual and the generated data. By comprehensively representing the values of all design parameters on the polar axes, we visualize the tendency of the whole parameters in four different phases. The normalized 989 samples, i.e., 250 SS, 267 SS + IM, 307 IM, and 165 AM, are presented in Fig. 9 (a), while Fig. 9 (b) shows 200 generated data for each phase. From Fig. 9 (a), the actual samples for the SS and SS + IM phases have similar spread patterns considering the entire design parameter, while the IM phase shows largely different patterns, as previously confirmed via the embedded distributions among the phases in Section 2. As shown in Fig. 9 (b), the patterns of the samples produced by our trained conditional GAN model realistically emulate ones of the existing data in Fig. 9 (a). It is worth noting that those generated data are a completely different set of data from the existing data, produced given the overall distribution of 13 design parameters rather than randomly extracted from pre-defined regions. Moreover, we find that the samples corresponding to a specific phase can be created by guiding the phase information, i.e., class labels of the four HEA phases, in our generative model.

Data augmentation using the generative neural network can provide the phase prediction model with additional information. For example, Fig. 10 (a) shows a scatter plot between the atomic size difference δ and the average mixing enthalpy ΔH_{mix} for the existing SS and SS + IM cases. While the dashed lines indicate the parametric approach for classifying SS and SS + IM phases, i.e., $6.7\% \delta$ and $-11.66 \text{ kJ/mol } \Delta H_{mix}$ [20,51–53], it is evident that the δ - ΔH_{mix} data of the two phases considerably overlap, particularly in the shaded region $3\% < \delta < 7.5\%$ and $-15 \text{ kJ/mol} < \Delta H_{mix} < 0 \text{ kJ/mol}$ from our dataset. In keeping with Fig. 1 and Fig. 9 (a), this high similarity between SS and SS + IM phases prevents the model from accurately predicting the phases. As represented in Fig. 10

Table 5

Performance comparison among several simpler classification algorithms using machine learning methods, i.e., SVM, decision tree, and XGBoost, and the proposed DNN model with hyper-parameter set λ^* .

Prediction model	Method	Average test accuracy
SVM	Machine learning-based	64.49%
Decision tree	Machine learning-based	71.78%
XGBoost	Machine learning-based	72.06%
Proposed DNN with λ^*	Deep learning-based	84.75%

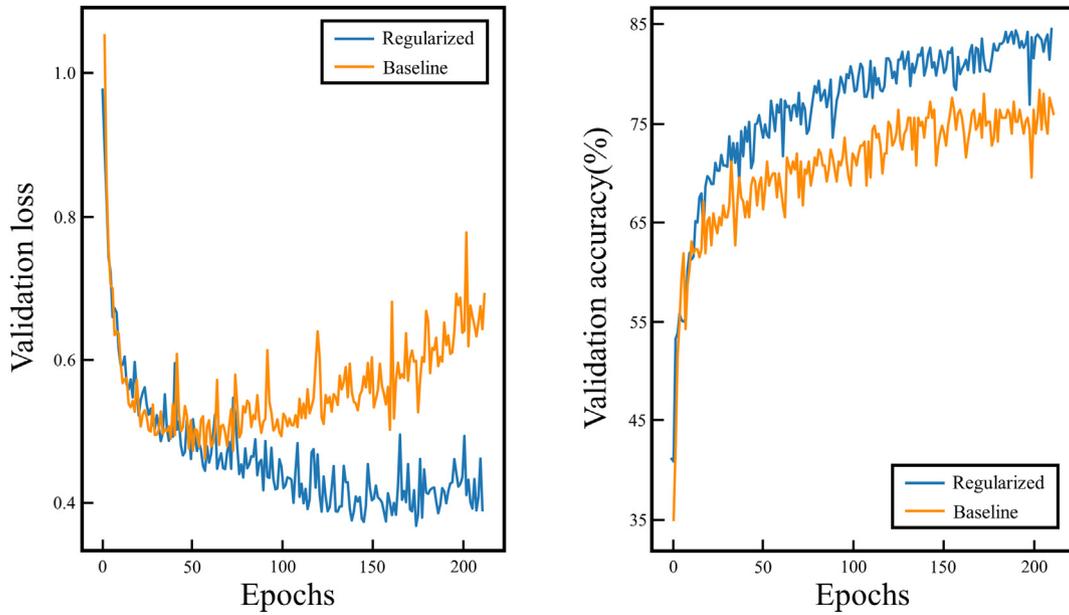


Fig. 7. Validation loss and accuracy comparison between the proposed model (Regularized) and the baseline model (Baseline) to determine the impact of the regularization methods: The proposed model refers to the DNN model applying the proposed hyper-parameter set λ^* in Table 4, while the baseline model is the same model as the proposed model but without employing two regularization methods, i.e., L2 regularization and dropout.

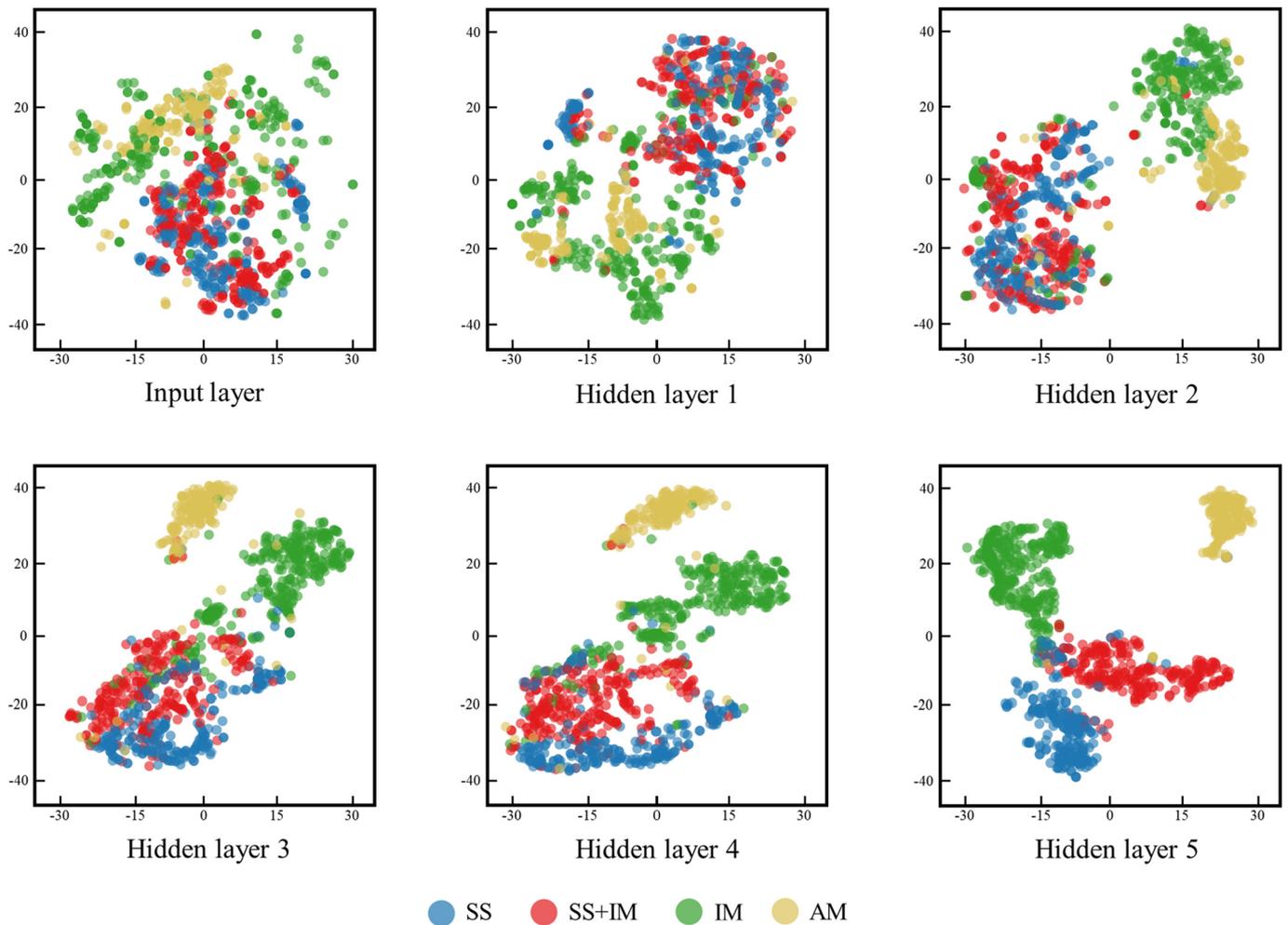


Fig. 8. Feature visualization of the proposed DNN model using t-SNE method: embedded feature distributions in the input layer and five successive hidden layers. The feature representations of four HEA phases become more disentangled as the input pass through the deeper hidden layers of our trained DNN model.

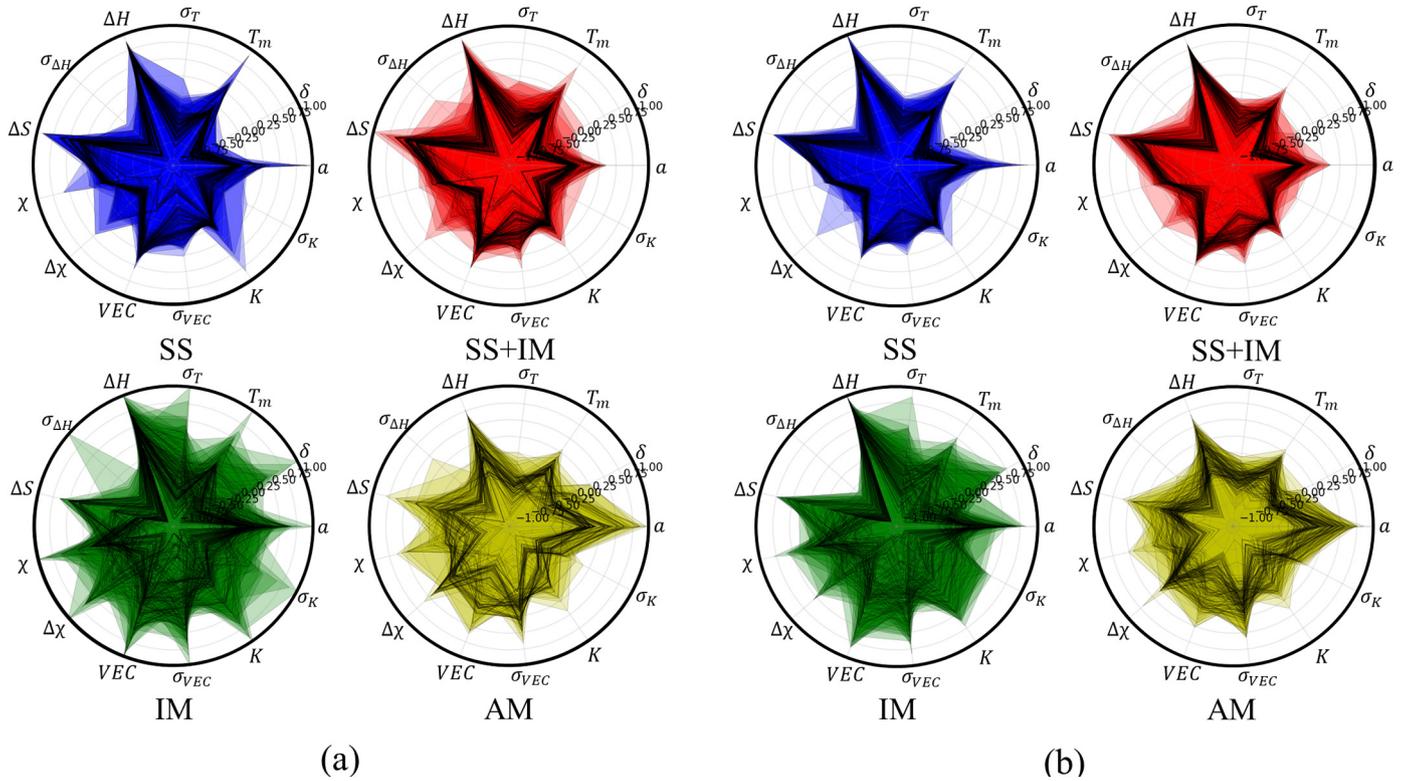


Fig. 9. Polar charts visualizing the distributed characteristics of the features in four phases: (a) plotting the actual values of 250 SS, 267 SS + IM, 307 IM, and 165 AM, respectively (b) plotting the generated values of 200 samples for each phase. It is shown that the HEA samples for the desired phases are successfully generated by combining the conditioning variables of each phase. Note that the generated samples are extracted from the trained model distribution that approximates the data distribution, while they are completely different data from the existing ones.

(a), we find that this ambiguous feature representation between the two phases leads to several misclassification cases by our DNN model, such as $\text{Al}_{0.08}\text{CoCrFeMnNi}$, $\text{CrMnFe}_{1.5}\text{Ni}_{0.5}\text{Al}_{1.2}$, $\text{Al}_{0.5}\text{CoCrCuFeNiTi}_{0.4}$, and $\text{CuCoNiCrAl}_{1.8}\text{Fe}$. On the other hand, Fig. 10 (b) shows δ versus ΔH_{mix} plot of the existing SS data and 200 SS samples generated by the conditional GAN model. It can be seen that the generated SS distribution generally approximates the actual data distribution in terms of δ and ΔH_{mix} parameters. Besides, it is observed that quite a few generated data belong to the shaded region that contains those misclassified cases, indicating that these additional data can be utilized to enrich the information for phase estimation. Thus, we anticipate that the proposed generation and augmentation approach can enhance the prediction capability of the neural network.

In order to evaluate the influence of data augmentation on phase prediction performance, quantitative results of several DNN models depending on the amount of the augmented samples are described in Table 6. We compare the DNN model that utilizes additional samples augmented by the conditional GAN (denoted as a DNN-Augment) with the model optimized via the Bayesian optimization in the previous section (denoted as a DNN-BO). Each augmented model is constructed and trained by gradually increasing the number of augmented data per phase, while the generated samples are attached only to the training datasets in 4-fold cross-validation procedures. As shown in Table 6, the DNN-Augment models with additional samples in general outperform the DNN-BO model merely using the existing data in terms of the predictive accuracy. While the predictive results of the DNN-BO model, i.e., 84.75%, slightly rises by 1.66% when 25 samples per phase are augmented, we find that the performance has a substantial increase by 5.54% from 50 augmented samples for each phase. It is observed that the more the neural network leverages the generated data, the more performance is improved, where the proposed model using 150 augmented samples per phase (DNN-Augment150) shows the highest accuracy by

93.17% in our experiments. Intriguingly, we also find that the performance with 200 more data somewhat diminishes than the results of DNN-Augment100 and DNN-Augment150. Considering that the existing training dataset used for each fold is comprised of around 750 samples, this implies that augmentation overtaking the number of the existing dataset can induce the neural network to be fitted excessively by the generated samples. Also, the results signify the importance of the number of augmented samples for model's performance improvement.

In addition, Fig. 11 describes confusion matrices of DNN-BO and DNN-Augment150 for delving into how these models differ in estimating per-phase prediction result. As shown in Fig. 11 (a), we observe that the DNN-BO model generally shows mediocre per-phase accuracy except AM phase, e.g., 77% and 81% accuracy for estimating SS and SS + IM phases, respectively. On the other hand, we observe that the DNN-Augment150 using conditional GAN-based data augmentation not only increases the total accuracy but also improves phase-specific performance, particularly for distinguishing SS, SS + IM, and IM phases. For instance, while the DNN-BO is more confused in classifying SS and SS + IM, showing around 16% actual SS samples and 15% actual SS + IM samples are wrongly estimated to SS + IM and SS, respectively, we find that the DNN-Augment150 achieves more precise per-phase estimation, showing no more than 8% error in distinguishing SS and SS + IM.

Finally, we validate our enhanced phase prediction model, i.e., DNN-Augment150, with additional HEA samples. Table 7 summarizes the validation results with several new HEAs, showing their actual phases, our model's predicted results, as well as softmax output for each phase. It is worth mentioning that we consider five additional HEAs that have been recently reported by several pieces of studies [54–57], where they are not considered in our dataset. As shown in Table 7, it is observed that our prediction model succeeds in estimating the whole five validation HEAs. While the softmax output can be interpreted as the prediction model's probability for estimating each phase, in particular, we find

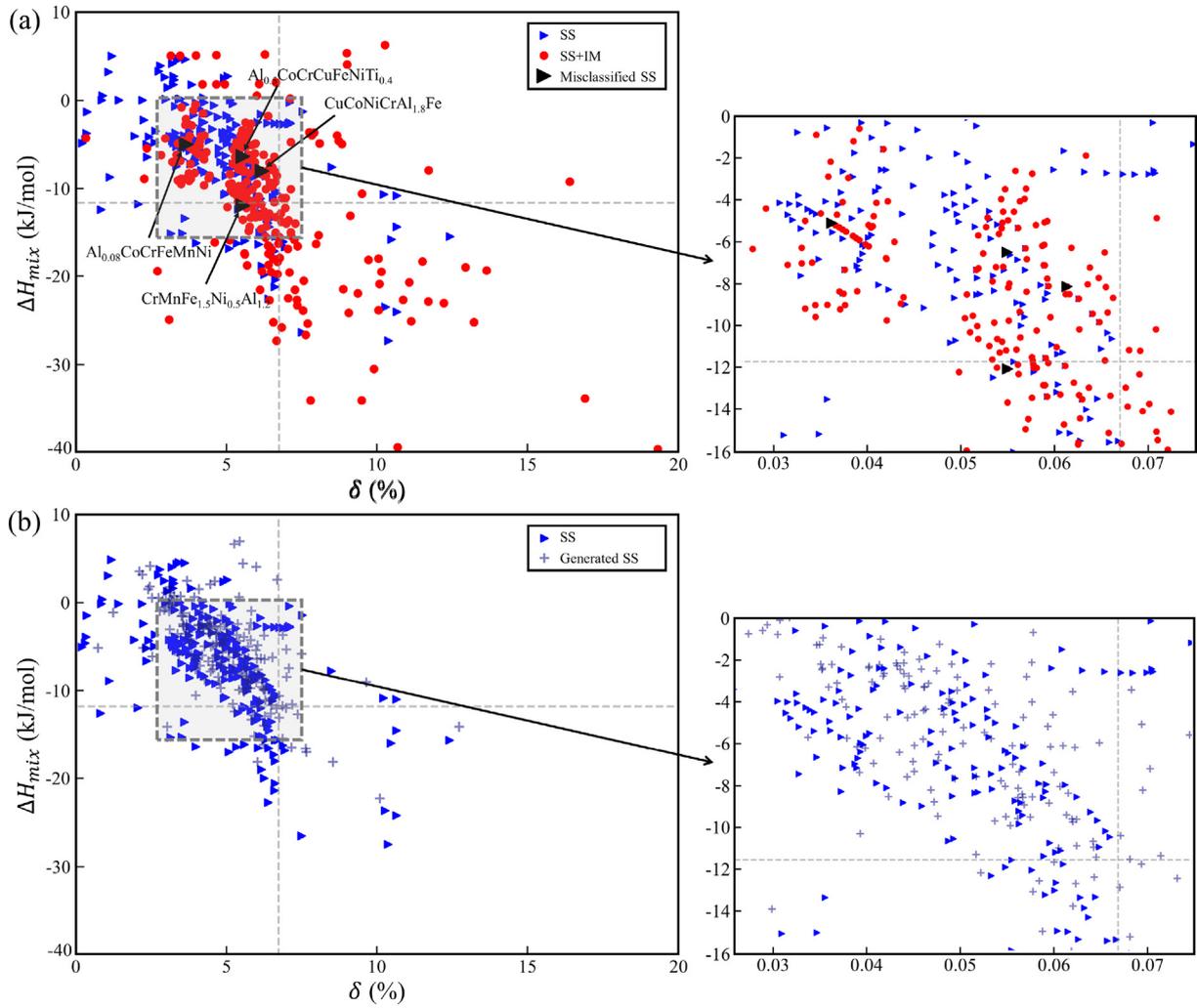


Fig. 10. Scatter plots of atomic size difference δ versus average mixing enthalpy ΔH_{mix} : (a) a plot for actual samples corresponding to SS and SS + IM phases, (b) a plot for actual SS and generated SS data. Dashed vertical and horizontal lines represent the parametric criteria for classifying SS and SS + IM phases, 6.7% δ and -11.66 kJ/mol ΔH_{mix} , respectively. The shaded box indicates the region ($3\% < \delta < 7.5\%$ and -15 kJ/mol $< \Delta H_{mix} < 0$ kJ/mol) where the samples of SS and SS + IM phases significantly overlap from our dataset.

that our prediction model correctly gives the prediction results with high confidence. For instance, our model has a 90.4% possibility for predicting $Fe_{4.5}Co_3CrVMn_{0.5}$ with SS phase, where there is only 9.5% for misjudging $Fe_{4.5}Co_3CrVMn_{0.5}$ as SS + IM and even less probability for estimating it as IM or AM phases. To sum up, a promising result for enhancing the predictive performance in the HEA system is obtained by our proposed approach with the generative framework and its data augmentation, showing the possibility of utilizing our proposed approach to assess future potential HEAs with high accuracy.

Table 6

Performance comparison among the DNN-based phase prediction models depending on the amount of the augmented samples by conditional GAN model. We examine the impact of the amount of augmented samples on predictive performance, while the samples generated by our conditional GAN model are only included in the training dataset.

Prediction model	Augmented samples per phase	Average test accuracy
DNN-BO	-	84.75%
DNN-Augment25	25	86.41%
DNN-Augment50	50	90.29%
DNN-Augment100	100	91.78%
DNN-Augment150	150	93.17%
DNN-Augment200	200	91.34%

Average test accuracy in bold represents the highest test accuracy among several prediction models considered in this table.

3.4. Significance of design parameters

We also attempt to assess the significance of each design parameter in our DL-based phase prediction. In order to examine how each parameter contributes to the estimation of the DNN model, we employ the layer-wise relevance propagation (LRP) method [58], which capitalizes on redistributing the output scores in order to find an appropriate explanation of the given prediction results. Fig. 12 describes the procedure of the relevance propagation in the DNN model. Unlike the forward propagation method, the LRP works by propagating the prediction results in the rear-most layer to the front-most one. This LRP technique considers that each hidden neuron has a certain confidence in a particular prediction. This confidence, so-called the relevance score, is repeatedly calculated based on the trained weights that connect neurons to others and the activated values of the previous layer:

$$R_i^{(l-1)} = \sum_j \frac{a_i^{(l-1)} w_{ij}^{(l)}}{\sum_i a_i^{(l-1)} w_{ij}^{(l)}} R_j^{(l)} \quad (6)$$

$R_i^{(l-1)}$ represents the relevance score of i^{th} neuron in the former layer, determined by the activation values $a_i^{(l-1)}$ in the previous layer, connections between the layers $w_{ij}^{(l)}$, and the weighted sum of the relevance scores in the rear layer. As the output scores spread to the input

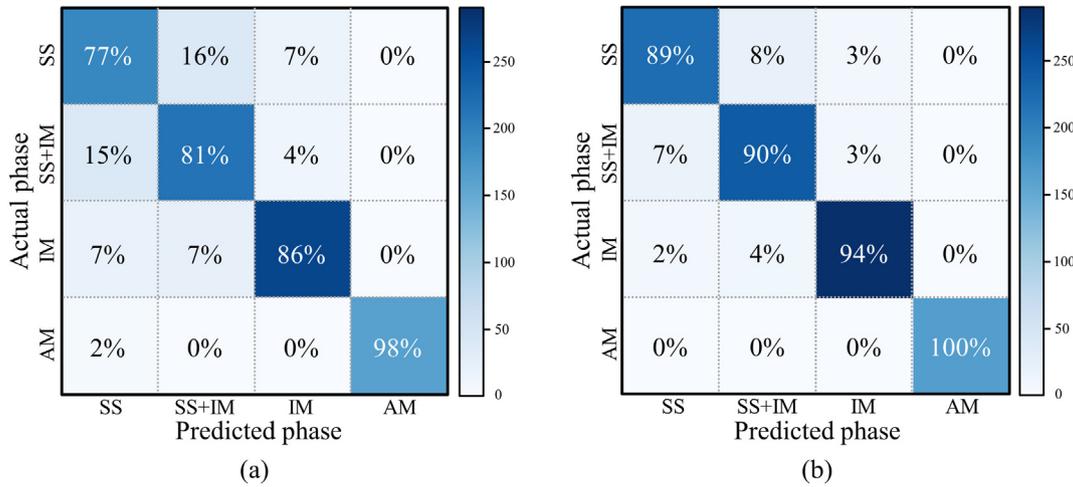


Fig. 11. Confusion matrices showing the phase-specific prediction results of two models: (a) a confusion matrix of DNN-BO (b) a confusion matrix of DNN-Augment150. Each confusion matrix is comprehensively calculated considering a total of 4-fold cross-validation results.

Table 7

DNN-Augment150 model's predicted results and their softmax output regarding several new validation HEA samples that have been recently discovered.

Validation HEAs	Actual phase	Predicted phase	Softmax output			
			SS	SS + IM	IM	AM
Fe _{4.5} Co ₃ CrVMn _{0.5} [54]	SS	SS	9.04e-1	9.50e-2	1.04e-5	4.14e-5
CoCu ₂ Mn ₃ Ni ₄ [55]	SS	SS	9.99e-1	5.70e-4	8.43e-5	2.21e-4
Al _{0.75} Cu _{3.0833} Fe _{3.0833} Mn _{3.0833} [56]	SS	SS	8.76e-1	1.23e-1	2.03e-4	5.04e-5
Cu _{3.33} Fe _{3.33} Mn _{3.33} [56]	SS	SS	9.99e-1	5.15e-5	8.98e-6	2.95e-6
Co _{1.75} Cr _{1.25} Fe _{5.5} NiMo _{0.3} C _{0.2} [57]	SS + IM	SS + IM	2.45e-2	9.74e-1	6.87e-4	5.19e-5

Softmax output in bold represent the highest softmax probabilities that the DNN-Augment150 yield among four phases, where the model's prediction results are determined.

layer through this successive redistribution process, the impact of each input feature (R_{input}) can be identified. By interpreting the impact of each input feature obtained by the trained weights from an already developed network, we analyze the importance of each design parameter in predicting the phase of HEAs.

The relevance scores for the entire actual samples (989 samples from our dataset) are obtained via our enhanced model in the previous section, i.e., DNN-Augment150, where we focus on analyzing the

feature significance for classifying the SS phase as an example. Note that the feature significance results for SS + IM, IM, and AM phases are presented in Supplementary Fig. 1–3. Fig. 13 represents the relevance scores of 13 design parameters for the SS phase, averaged over the whole 989 HEA samples. The value of relevance score means which design parameters are significantly used for the phase prediction, rather than meaning the actual magnitude of each parameter. We observe that the most important parameters for predicting the SS phase are

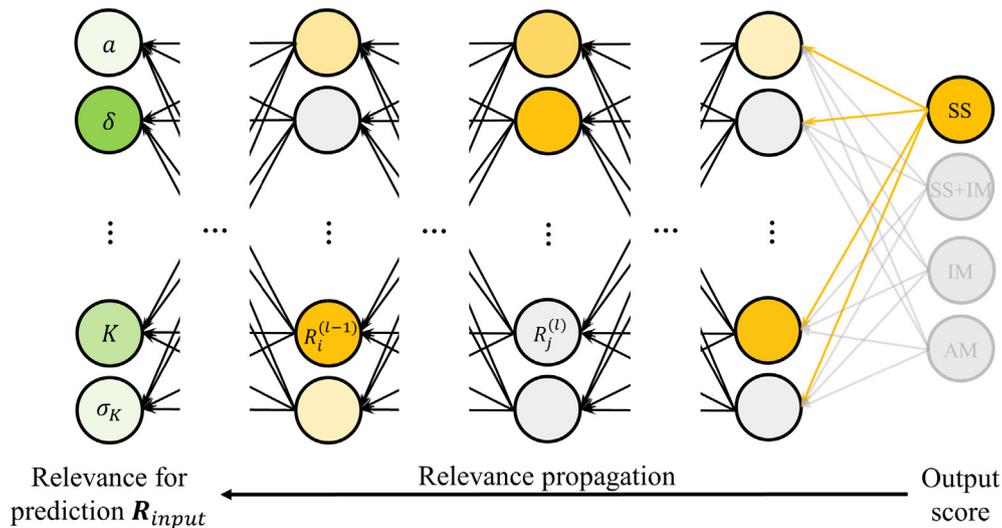


Fig. 12. Schematic diagram of the relevance propagation process for examining the significance of the design parameters (R_{input}) for the DL-based phase prediction. Based on the trained weights inside the DNN model, the relevance score can be calculated by propagating the predictive output score to the input layer.

acquired in the order of ΔH_{mix} , δ , and ΔS_{mix} , which indicates that those parameters should be prioritized when designing the SS phase of HEAs. More specifically, the most significant parameters for the SS phase that are calculated our enhanced DNN model and the LRP method indicates that those parameters should be preferentially considered to discover HEAs of the SS phase, owing to the fact that more relevance score of a particular phase represents more attention of the DNN model to estimate a sample in a particular phase.

It is found that the significant design parameters deduced by the LRP method generally coincide with those dictated by the existing parametric rules for the formation of the SS phase. For instance, ΔH_{mix} , which is the variable considering binary interactions in a liquid phase, was statistically inferred as the influential feature for making the SS phase [16,49]. In line with the Hume-Rothery rule, it is also known that δ plays a vital role in forming the SS phase [16,59,60]. Besides, Yeh et al. [61] proposed that ΔS_{mix} is the driving force behind the formation of a random SS.

In order to understand the behavior of δ , ΔH_{mix} , and ΔS_{mix} , we further study how the level of importance for those parameters evolves as training of the prediction model progresses. While Fig. 14 (a) represents the learning curve of our proposed model showing the accuracy during the training process, we visualize in Fig. 14 (b) how the relevance scores of those parameters change simultaneously. Interestingly, it can be seen that the relative significance among δ , ΔH_{mix} , and ΔS_{mix} changes as learning continues and the predictive accuracy of the model increases. At the early stage of training where the model shows less than 90% accuracy, δ seems to have the largest impact subsequently followed by ΔS_{mix} and ΔH_{mix} . The relevance score of ΔH_{mix} passes over that of ΔS_{mix} as our DNN achieves approximately 93–95% accuracy. Overall, while the relevance score of ΔS_{mix} and δ appear to decline gradually, that of ΔH_{mix} continuously increases and becomes the most significant parameter when the learning is completed. This observation from Fig. 14 (a) and (b) possibly suggests that the relative significance among the design parameters continuously evolves in such a way that the prediction accuracy of the DNN model improves.

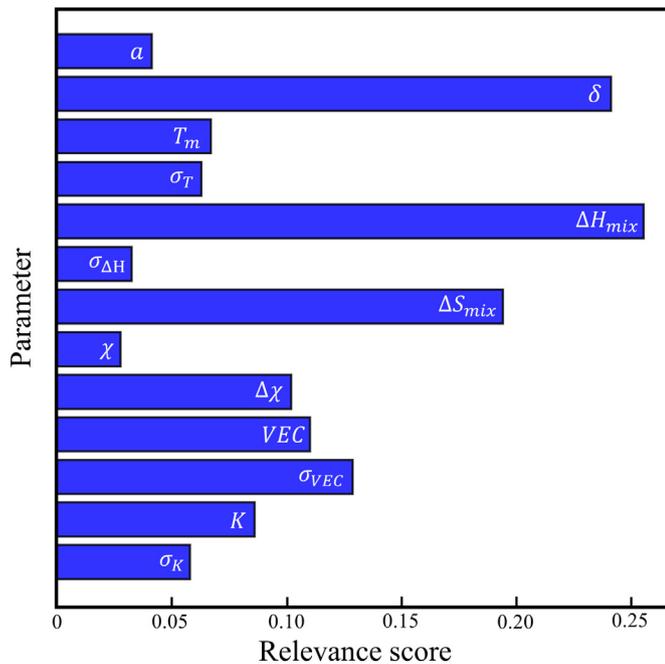


Fig. 13. Bar chart showing the relevance scores of 13 design parameters in predicting the SS phase of HEAs. By interpreting a black-box DNN model with LRP method, it is observed that ΔH_{mix} , δ , and ΔS_{mix} are the significant design parameters for estimating the SS phase.

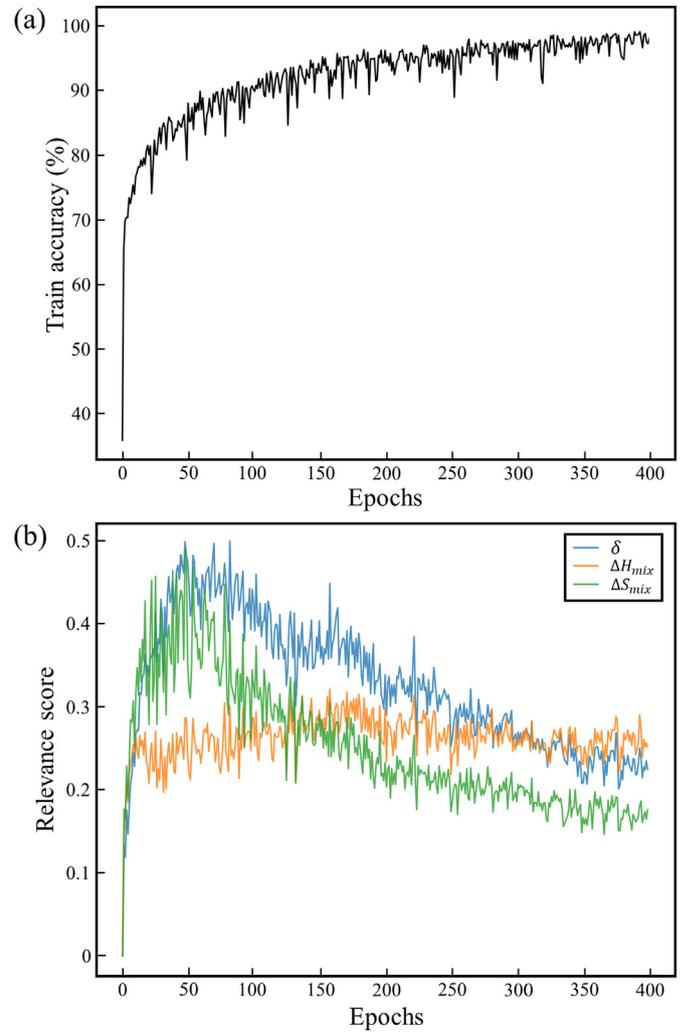


Fig. 14. Behaviors over training epochs of the model performance and the significance of design parameters for predicting SS phase: (a) learning curve showing the training accuracy as learning progresses (b) changes in the relative significance of δ , ΔH_{mix} , ΔS_{mix} in prediction of the SS phase.

4. Conclusions

In this study, we proposed a systematic deep learning approach for predicting the four phases of HEAs with high accuracy. We developed our deep learning methodology from perspectives of optimization, generation, and explanation. The main contributions of this work are summarized as follows.

- Firstly, we found that improved results for phase prediction can be obtained by hyper-parameter optimization. Through the Bayesian optimization process for overall settings related to model architecture, training, and regularization, we attained the regularized DNN model with five hidden layers, which achieves 84.75% testing accuracy. Besides, we demonstrated the distinguishing tendency among the four phases via visualization of high-dimensional features, and observed that the feature distributions of all the phases are distinctively separated in the latter part of the hidden layers of our trained DNN model.
- Secondly, we utilized a generative design of the neural network for producing realistic samples that approximate data distribution of the existing HEA samples. In particular, we verified that it is possible to generate the data for a specific phase by conditioning the phase information to the generative adversarial framework. By examining the influence of augmentation on the model performance, we established

a phase prediction model with unprecedented 93.17% testing accuracy for simultaneously estimating the four phases of HEAs.

- Lastly, based on the interpretation of our phase prediction model, the significant design parameters for predicting the SS phase were explained. By understanding the decision process of the deep neural network, we found that the parameters in the order of ΔH_{mix} , δ , and ΔS_{mix} have a high significance in classifying the SS phase, consistent with the existing parametric rules.

Regarding future research directions, we mainly aim at discovering novel desired HEAs via deep learning-based phase prediction model as established in this study, as well as actual verification with several experiments. Moreover, designing additional features for phase prediction of HEAs is valuable to be further investigated to maximize the DNN model's phase prediction performance.

Data availability

The data used in this manuscript are available from the corresponding author upon reasonable request.

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.

Acknowledgment

The research of S.L and S.Y.L was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government Ministry of Science and ICT (MSIT) (No. 2020R1A2C1009744), and in part by the High-Potential Individuals Global Training Program of Institute for Information and Communications Technology Promotion (IITP) under Grant 2019-0-01589. The research of H.J and S.-B was supported in part by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean Government (MOTIE) (20188550000290, the Development of Meta-Silicide Thermoelectric Semiconductor and Metrology Standardization Technology of Thermoelectric Power Module), and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2020R1C1C1004291).

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.matdes.2020.109260>.

References

- [1] A. Takeuchi, A. Inoue, Calculations of mixing enthalpy and mismatch entropy for ternary amorphous alloys, *Mater. Trans.* 41 (2000) 1372–1378.
- [2] W.L. Johnson, Fundamental aspects of bulk metallic glass formation in multicomponent alloys, *Materials Science Forum*, vol. 225, Trans Tech Publ 1996, pp. 35–50.
- [3] M.C. Gao, J.-W. Yeh, P.K. Liaw, Y. Zhang, *High-Entropy Alloys: Fundamentals and Applications*, Springer, 2016.
- [4] M.-H. Tsai, J.-W. Yeh, High-entropy alloys: a critical review, *Mater. Res. Lett.* 2 (2014) 107–123.
- [5] M.-H. Chuang, M.-H. Tsai, W.-R. Wang, S.-J. Lin, J.-W. Yeh, Microstructure and wear behavior of Al_xCo_{1-x}5CrFeNi_{1.5}Ti_y high-entropy alloys, *Acta Mater.* 59 (2011) 6308–6317.
- [6] C.-C. Juan, K.-K. Tseng, W.-L. Hsu, M.-H. Tsai, C.-W. Tsai, C.-M. Lin, S.-K. Chen, S.-J. Lin, J.-W. Yeh, Solution strengthening of ductile refractory HfMoxNbTaTiZr high-entropy alloys, *Mater. Lett.* 175 (2016) 284–287.
- [7] X.-W. Qiu, Y.-P. Zhang, L. He, C.-g. Liu, Microstructure and corrosion resistance of AlCrFeCuCo high entropy alloy, *J. Alloys Compd.* 549 (2013) 195–199.
- [8] C.-J. Tong, M.-R. Chen, J.-W. Yeh, S.-J. Lin, S.-K. Chen, T.-T. Shun, S.-Y. Chang, Mechanical performance of the Al_xCo_{1-x}CrCuFeNi high-entropy alloy system with multiprincipal elements, *Metall. Mater. Trans. A* 36 (2005) 1263–1271.
- [9] C. Li, J. Li, M. Zhao, Q. Jiang, Effect of alloying elements on microstructure and properties of multiprincipal elements high-entropy alloys, *J. Alloys Compd.* 475 (2009) 752–757.
- [10] S. Zhai, J. Rojas, N. Ahlborg, K. Lim, M.F. Toney, H. Jin, W.C. Chueh, A. Majumdar, The use of poly-cation oxides to lower the temperature of two-step thermochemical water splitting, *Energy Environ. Sci.* 11 (2018) 2172–2178.
- [11] Z. Fan, H. Wang, Y. Wu, X. Liu, Z. Lu, Thermoelectric performance of PbSnTeSe high-entropy alloys, *Mater. Res. Lett.* 5 (2017) 187–194.
- [12] M.-H. Tsai, Three strategies for the design of advanced high-entropy alloys, *Entropy* 18 (2016) 252.
- [13] O. Senkov, D. Miracle, Effect of the atomic size distribution on glass forming ability of amorphous metallic alloys, *Mater. Res. Bull.* 36 (2001) 2183–2198.
- [14] Q. Hu, S. Guo, J. Wang, Y. Yan, S. Chen, D. Lu, K. Liu, J. Zou, X. Zeng, Parametric study of amorphous high-entropy alloys formation from two new perspectives: atomic radius modification and crystalline structure of alloying elements, *Sci. Rep.* 7 (2017) 1–12.
- [15] M.-H. Tsai, R.-C. Tsai, T. Chang, W.-F. Huang, Intermetallic phases in high-entropy alloys: statistical analysis of their prevalence and structural inheritance, *Metals* 9 (2019) 247.
- [16] Y. Zhang, Y.J. Zhou, J.P. Lin, G.L. Chen, P.K. Liaw, Solid-solution phase formation rules for multi-component alloys, *Adv. Eng. Mater.* 10 (2008) 534–538.
- [17] S. Guo, C. Ng, J. Lu, C. Liu, Effect of valence electron concentration on stability of fcc or bcc phase in high entropy alloys, *J. Appl. Phys.* 109 (2011) 103505.
- [18] R. Feng, P.K. Liaw, M.C. Gao, M. Widom, First-principles prediction of high-entropy-alloy stability, *Npj Comput. Mater.* 3 (2017) 1–7.
- [19] C. Wen, Y. Zhang, C. Wang, D. Xue, Y. Bai, S. Antonov, L. Dai, T. Lookman, Y. Su, Machine learning assisted design of high entropy alloys with desired property, *Acta Mater.* 170 (2019) 109–117.
- [20] W. Huang, P. Martin, H.L. Zhuang, Machine-learning phase prediction of high-entropy alloys, *Acta Mater.* 169 (2019) 225–236.
- [21] Z. Zhou, Y. Zhou, Q. He, Z. Ding, F. Li, Y. Yang, Machine learning guided appraisal and exploration of phase design for high entropy alloys, *npj Comput. Mater.* 5 (2019) 1–9.
- [22] M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, *J. Big Data* 2 (2015) 1.
- [23] D.B. Miracle, O.N. Senkov, A critical review of high entropy alloys and related concepts, *Acta Mater.* 122 (2017) 448–511.
- [24] J.-W. Yeh, Physical metallurgy of high-entropy alloys, *JOM* 67 (2015) 2254–2261.
- [25] Y. Dong, Y. Lu, L. Jiang, T. Wang, T. Li, Effects of electro-negativity on the stability of topologically close-packed phase in high entropy alloys, *Intermetallics* 52 (2014) 105–109.
- [26] L.V.D. Maaten, G. Hinton, Visualizing data using t-sne, *J. Mach. Learn. Res.* 9 (2008) 2579–2605.
- [27] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, *Proceedings of the 27th International Conference on Machine Learning (ICML) 2010*, pp. 807–814.
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* 15 (2014) 1929–1958.
- [29] E. Phaisangittisagul, An Analysis of the Regularization between L2 and Dropout in Single Hidden Layer Neural Network, in: 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), IEEE, 2016 174–179.
- [30] B. Neyshabur, S. Bhojanapalli, D. McAllester, N. Srebro, Exploring generalization in deep learning, *Advances in Neural Information Processing Systems 2017*, pp. 5947–5956.
- [31] F. He, T. Liu, D. Tao, Control batch size and learning rate to generalize well: Theoretical and empirical evidence, *Advances in Neural Information Processing Systems 2019*, pp. 1141–1150.
- [32] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. De Freitas, Taking the human out of the loop: a review of bayesian optimization, *Proc. IEEE* 104 (2015) 148–175.
- [33] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, *J. Mach. Learn. Res.* 13 (2012) 281–305.
- [34] J.S. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, *Advances in Neural Information Processing Systems 2011*, pp. 2546–2554.
- [35] J. Wilson, F. Hutter, M. Deisenroth, Maximizing acquisition functions for bayesian optimization, *Advances in Neural Information Processing Systems 2018*, pp. 9884–9895.
- [36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, *Advances in Neural Information Processing Systems 2014*, pp. 2672–2680.
- [37] J. Schmidt, M.R. Marques, S. Botti, M.A. Marques, Recent advances and applications of machine learning in solid-state materials science, *Npj Comput. Mater.* 5 (2019) 1–36.
- [38] A. Ghahramani, F.M. Watt, N.M. Luscombe, Generative adversarial networks uncover epidermal regulators and predict single cell perturbations, *bioRxiv* (2018) 262501.
- [39] X. Li, Z. Yang, L.C. Brinson, A. Choudhary, A. Agrawal, W. Chen, A Deep Adversarial Learning Methodology for Designing Microstructural Material Systems, in: ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, 2018.
- [40] L. Banko, Y. Lysoogorskiy, D. Grochla, D. Naujoks, R. Drautz, A. Ludwig, Predicting structure zone diagrams for thin film synthesis by generative machine learning, *Comms. Mater.* 1 (2020) 1–10.

- [41] A. Iyer, B. Dey, A. Dasgupta, W. Chen, A. Chakraborty, A conditional generative model for predicting material microstructures from processing methods, arXiv (2019) preprint arXiv:1910.02133.
- [42] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv (2014) preprint arXiv:1411.1784.
- [43] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv (2015) preprint arXiv:1502.03167.
- [44] S. Santurkar, D. Tsipras, A. Ilyas, A. Madry, How does batch normalization help optimization? Advances in Neural Information Processing Systems 2018, pp. 2483–2493.
- [45] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, Proceedings of the 30th International Conference on Machine Learning (ICML) 2013, p. 3.
- [46] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, Advances in Neural Information Processing Systems 2016, pp. 2234–2242.
- [47] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv (2014) preprint arXiv:1412.6980.
- [48] A. Takeuchi, A. Inoue, Quantitative evaluation of critical cooling rate for metallic glasses, Mater. Sci. Eng. A 304 (2001) 446–451.
- [49] G. Sheng, C.T. Liu, Phase stability in high entropy alloys: formation of solid-solution phase or amorphous phase, Pro. Nat. Sci-Mater. 21 (2011) 433–446.
- [50] X. Yang, Y. Zhang, Prediction of high-entropy stabilized solid-solution in multi-component alloys, Mater. Chem. Phys. 132 (2012) 233–238.
- [51] Y. Zhang, T.T. Zuo, Z. Tang, M.C. Gao, K.A. Dahmen, P.K. Liaw, Z.P. Lu, Microstructures and properties of high-entropy alloys, Prog. Mater. Sci. 61 (2014) 1–93.
- [52] Y. Ye, Q. Wang, J. Lu, C. Liu, Y. Yang, High-entropy alloy: challenges and prospects, Mater. Today 19 (2016) 349–362.
- [53] S. Guo, Q. Hu, C. Ng, C. Liu, More than entropy in high-entropy alloys: forming solid solutions or amorphous phase, Intermetallics 41 (2013) 96–103.
- [54] J. Yang, Y.H. Jo, D.W. Kim, W.-M. Choi, H.S. Kim, B.-J. Lee, S.S. Sohn, S. Lee, Effects of transformation-induced plasticity (TRIP) on tensile property improvement of Fe₄₅Co₃₀Cr₁₀V₁₀Ni₅-xMnx high-entropy alloys, Mater. Sci. Eng. A 772 (2020) 138809.
- [55] D.G. Kim, Y.H. Jo, J.M. Park, W.-M. Choi, H.S. Kim, B.-J. Lee, S.S. Sohn, S. Lee, Effects of annealing temperature on microstructures and tensile properties of a single FCC phase CoCuMnNi high-entropy alloy, J. Alloys Compd. 812 (2020) 152111.
- [56] J. Moon, J.M. Park, J.W. Bae, H.-S. Do, B.-J. Lee, H.S. Kim, A new strategy for designing immiscible medium-entropy alloys with excellent tensile properties, Acta Mater. 193 (2020) 71–82.
- [57] H. Kwon, J. Moon, J.W. Bae, J.M. Park, S. Son, H.-S. Do, B.-J. Lee, H.S. Kim, Precipitation-driven metastability engineering of carbon-doped CoCrFeNiMo medium-entropy alloys at cryogenic temperature, Scr. Mater. 188 (2020) 140–145.
- [58] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digit. Signal Process. 73 (2018) 1–15.
- [59] U. Mizutani, The hume-rothery rules for structurally complex alloy phases, Surface Properties And Engineering Of Complex Intermetallics, World Scientific 2010, pp. 323–399.
- [60] A. Takeuchi, A. Inoue, Classification of bulk metallic glasses by atomic size difference, heat of mixing and period of constituent elements and its application to characterization of the main alloying element, Mater. Trans. 46 (2005) 2817–2829.
- [61] J.-W. Yeh, S.-K. Chen, S.-J. Lin, J.-Y. Gan, T.-S. Chin, T.-T. Shun, C.-H. Tsau, S.-Y. Chang, Nanostructured high-entropy alloys with multiple principal elements: novel alloy design concepts and outcomes, Adv. Eng. Mater. 6 (2004) 299–303.