

# Evaluating generative models for inverse design of high-entropy refractory alloys

Yen Cheng Tung <sup>1</sup>, Arindam Debnath <sup>1</sup>, Wesley F. Reinhart <sup>1,2</sup>

1) Department of Materials Science and Engineering, Pennsylvania State University,  
University Park, PA 16802

2) Institute for Computational and Data Sciences, Pennsylvania State University,  
University Park, PA 16802

KEYWORDS: Machine learning, Deep learning, Generative model, High-Entropy Alloys

---

## 1 Abstract

Generative modeling is an innovative new method to design functional materials. In this article, we quantify the performance of several generative modeling architectures – autoencoder, variational autoencoder, and generative adversarial network – to design novel materials. These are compared to rational design method by case study of refractory high-entropy alloys for ultra-high-temperature applications. Furthermore, we apply a series of validation methods to evaluate the models' effectiveness and express the current difficulty. Overall, cAE is able to create versatile compositions and keep at lower similarity;

cVAE is capable of generating compositions with high accuracy; WcGAN has ability of creating novel compositions compared to training data. However, since the mode collapse phenomenon occurred in our results, inverse design still cannot totally replace rational design method at this stage.

## 2 Introduction

### 2.1 High entropy refractory alloys

High-Entropy Alloys (HEAs) are alloys comprised of 5 or more different principal elements (5-35%) and other minor elements (<5%), in contrast to conventional alloys, which are usually composed of 1 or 2 principal elements and minor percentages of the other elements<sup>2,6,24,31,32</sup>. With many different elements, the high mixing entropy can stabilize the phases in solid solution with simple crystal structures such as BCC, unlike conventional alloys, which usually contain intermetallic phases<sup>6,24,31,32,34</sup>. This unique structural feature grants the HEAs distinct and desirable characteristics, such as high toughness<sup>6,31,34</sup>, high-temperature strength<sup>6,26,31,32,34</sup>, good oxidation<sup>6,26,31,34</sup>, corrosion resistance<sup>6,31,32</sup>, and so forth.

The toughness of HEAs is generally higher than conventional alloys at elevated temperatures due to their excellent thermal resistance. Two aspects explain this property: (1) sluggish diffusion of atoms in the crystal<sup>6,19,24</sup> and (2) severe lattice-distortion effects<sup>6</sup>.

Compared to the conventional alloys structure, within the HEA structure, the types of atoms in each lattice sites are more likely to be different than neighboring atoms, it directly leads to the slower diffusion rate. Following concepts of vacancy, atoms are preferring to stay in the low-energy location and be "trapped", in opposition, atoms become unstable to stay at the higher energy site and jump back to its original site. Either phenomenon slows down the diffusion rate. Since the neighboring atoms in conventional alloys are mostly identical, atoms jump frequency is higher than HEAs'.

Applying many kinds of elements with different size leads the lattice distorted severely and raise the strain energy and Gibbs energy<sup>6,31,32</sup>. This helps the materials become stable under high temperature, since the raised Gibbs energy decreases the energy difference in defect-containing crystals, consequently reducing the driving force to eliminate defects. The distortion also interferes the propagation of phonons and electrons, so HEAs generally have lower electric and thermal conductivity<sup>24,31</sup>.

High temperature structural materials remain in high demand in several industries, and high entropy refractory alloys are great candidates. However, the speed of development of HEAs is slow due to the cost of raw materials and requirement of specialized high-temperature instruments. This results in relatively few examples in the published scientific literature and makes it difficult to predict properties of new HEAs. Generative models capable of inverse design therefore stand to greatly

accelerate the development process if they can be implemented effectively<sup>2,4,5,10,11,13,27</sup>.

## 2.2 Data-driven design

When designing functional materials, it is challenging to describe the relationship between chemical composition, internal structure, and macroscopic properties (e.g., mechanical properties) due to the multi-scale physical interactions that lead to these properties. Traditional material design methods heavily depend on scientific intuition by referring to historical experimental data and predicting new materials' possible properties by drawing analogies<sup>19,34</sup>. However, this sometimes leads experiments in the wrong direction such that significant time and effort must be invested to correctly establish the relationship among new materials and their properties and get back on track. As manufacturing processes become more complex such that they have a large number of independent input variables, designing new materials becomes more challenging and unpredictable than before.

New computation methods have been developed recently to solve complex science and engineering problems including material design. Traditionally, material science was focused on exploring the relation among different materials, measuring, and building logical connections between physical concepts to benefit experiments (e.g., through phenomenological models). The maturation of reliable simulation tools like

CALPHAD, which is widely used to evaluate stability of multicomponent systems, and first principles methods like density functional theory have matured in conjunction with rapidly growing computational capacity of hardware, data-driven rational design has become more practical<sup>9,16,22,35</sup>.

However, according to recent results,<sup>17</sup> first principles calculations still consume many computational resources compared to neural-network-based models. On the other hand, surrogate models based on neural networks can now reach the million-atoms scale with faster calculation rates under the same conditions and keep comparable accuracy. In fact, Deep Neural Network (DNN) models act as very capable surrogate models in several fields with their flexibility in recent years and have already been applied in different materials field and acquired recognition for their scientific capabilities<sup>17,25,29,35</sup>. DNN is common structure to solve regression or classification problems with structured data -- convolutional NN is a great candidate to analyze image data, while recurrent NN can detect trends in time-varying data.

In recent HEAs experiments<sup>25,36,37,38</sup>, some are focused on predicting properties such as stability and hardness through DNN models to acquire promising results. However, using the forward model sometimes is not straightforward enough to design effective HEA materials since it still depends on manipulating the inputs to find the desired output. With this approach it can be slow or impossible to find a

suitable input that produces the desired output, depending on the complexity of the problem.

Generative models have become a great candidate to develop new materials, including refractory HEAs, through direct approximation of the inverse function. Through manipulating a low-dimensional “latent space,” the model can directly infer a composition that should yield the desired properties (by estimating the conditional distribution). In the following, we compare the performance of the autoencoder (AE), variational autoencoder (VAE), and Generative Adversarial Network (GAN) model architectures<sup>4,5,8,10,27</sup>.

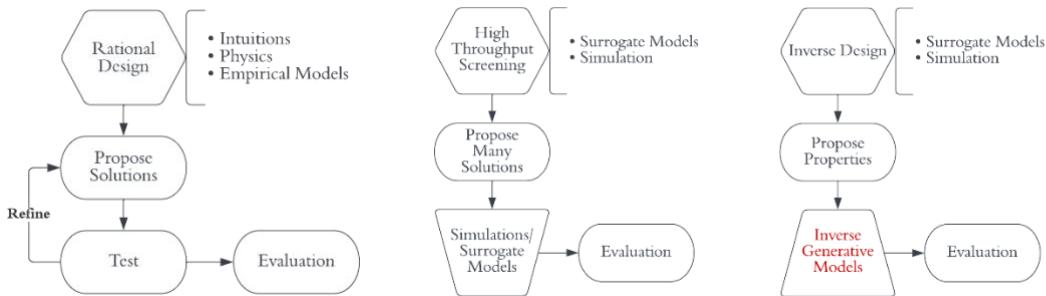


Figure 1. Schematic illustration of different material design schemes: (A) rational design, (B) high throughput screening, and (C) inverse design.

### 3 Materials and Methods

#### 3.1 Experimental data

The experimental data consists of 529 different high entropy refractory alloys compositions, with each observation being comprised of a vector with concentration of

22 different elements: Al, Be, Co, Cr, Cu, Fe, Hf, Ir, Mn, Mo, Nb, Ni, O, Re, Ru, Si, Ta, Ti, V, W, Y, Zr. The target properties are shear modulus and toughness. The distribution of the dataset is shown below (Fig. 2).



Figure 2. Scatter plot showing the distribution of toughness fracture and shear modulus in training data.

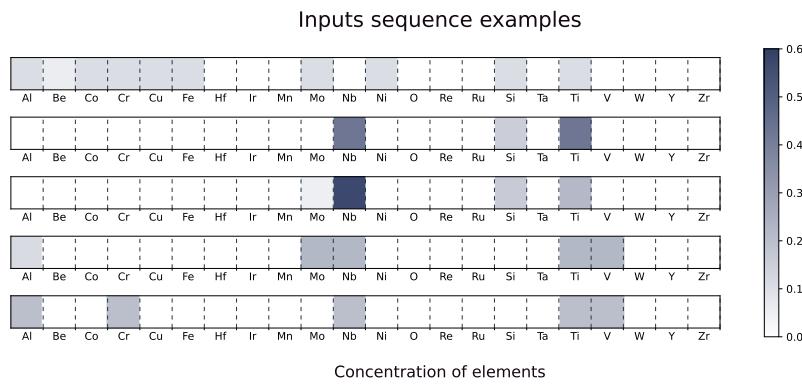


Figure 3. Five examples of input composition vectors.

### 3.2 Inverse model architectures

The main goal of training the generative model is to create unlimited novel HEA designs whose properties fit the requested input conditions (i.e., mechanical properties). For

example, if we select fracture toughness with 97.53 (MPa/m<sup>1/2</sup>) and shear modulus with 15.07 (GPa) as input conditions, the model should output an ensemble of compositions whose shear modulus and toughness closely match. There are many clear advantages of applying the model while designing novel materials, although inverse design has a limited ability to explore far outside the training domain. It is more intuitive compared to conventional rational design and can serve as a recommender system without forcing the scientist to select compositions manually and send them as input to a predictive model (even if that model is accurate, it will be tedious to do this many times).

### 3.3 Autoencoder architecture<sup>12,15,20</sup>

Autoencoder (AE) is an unsupervised learning algorithm with multi-layer neural networks that attempts to compress a high-dimensional input to a low-dimensional latent space and then reconstruct the input on the other side. It is trained by using a recirculation algorithm which is to compare its output to the original input, minimizing the reconstruction error. It usually formed with 3 parts: encoder  $f(x)$ , latent space  $h$ , and decoder  $g(x)$  (see Fig. 3-1). Encoder  $P(h/X)$  is generally responsible for extracting useful properties from higher dimensions to lower dimensions of hidden layers. Decoder  $Q(X/h)$  has the opposite function compared to encoder in that it reconstructs  $X$  from the latent space.

The goal of applying AE is not just doing copying from input to output, is trained to be able to use the decoder to reproduce a reconstruction with random latent code  $z$ . In essence, this allows it to learn abstract representations of the original distribution. If the model is well trained, this can be used to generate unlimited recipes of HEAs with the pretrained decoder (i.e., in inference mode).

The purpose of generative models is to increase the probability of  $X$  during the entire process, with the relation:

$$P(X) = \int P(X|z; \theta)P(z)dz$$

The idea behind the relation is achieving “maximum likelihood” between the training samples and reconstructed samples whose concept is like doing linear combination from the high dimension samples which is hard to calculate by conventional methods. If the model trains well, it will have ability to generate novel samples that appear to come from the same distribution of the training data.

We are not expecting that the model learns perfectly  $Q(P(X)) = X$ , since it will run the chance of overfitting and only copy the input data without learning to extract salient information to  $h$ . Reducing the capacity in encoder and decoder networks can mitigate the risk of overfitting.

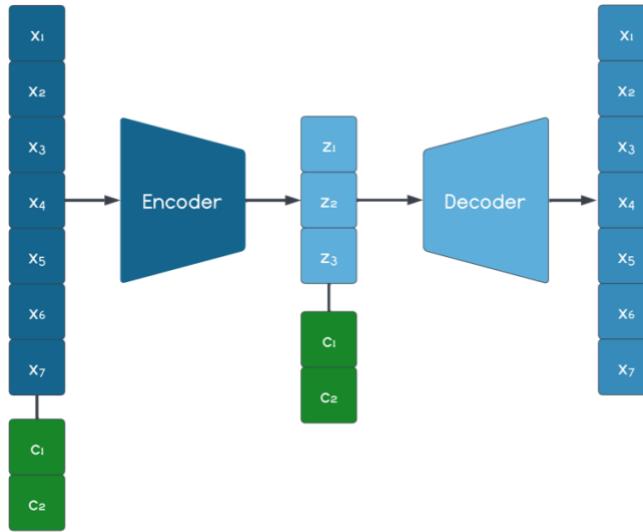


Figure 4. Schematic of a conditional autoencoder with 7 inputs, 2 conditions, and size-3 latent code.

### 3.4 Variational Autoencoder (VAE) architecture<sup>12,15,20</sup>

A more nuanced way to encode information in the latent space uses probabilistic  $P(X)$  mappings with a finite variance rather than assuming  $P(X)$  maps to exactly one point like in the autoencoder. This mitigates one of the most serious issues in autoencoders -- the regularity or smoothness of the latent space; the inverse mapping from two nearby points in latent space should not be too different. In practice, we enforce the distribution learned by the encoder closely resembles the Normal distribution. By adding noise to the data, the outputs become more accurate and stable interpolations between two points are obtained since the model has considered the overlapped tails of nearby distributions while training.

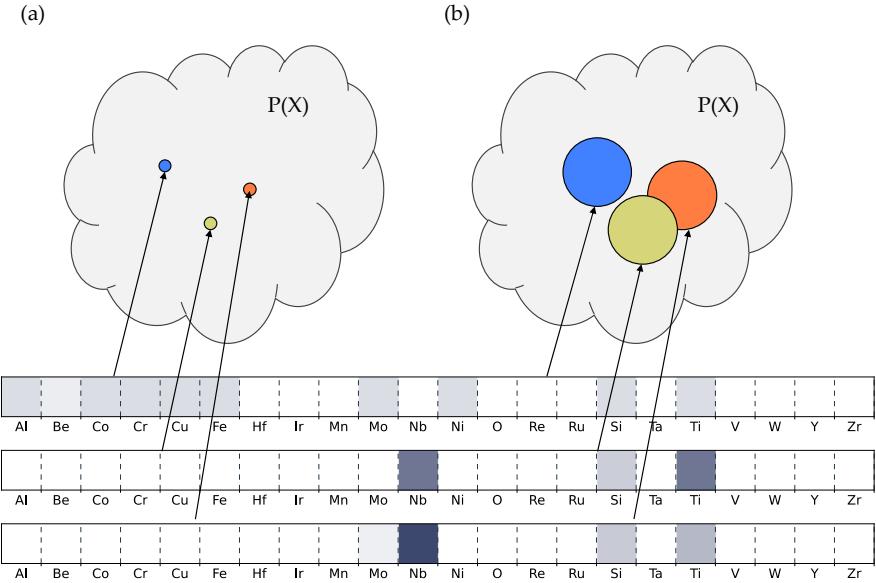


Figure 5. Schematic illustration of (a)  $P(X)$  in autoencoder (b)  $P(X)$  in variational autoencoder.

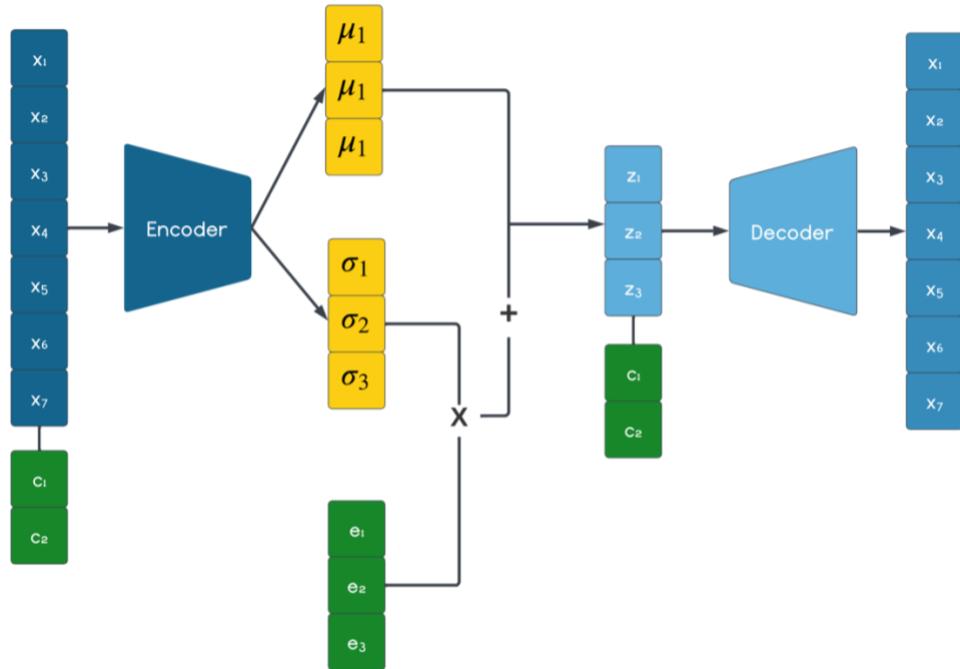


Figure 6. Schematic of the conditional Variational Autoencoder architecture.

The basic idea behind variational autoencoder is unchanged with autoencoder, which is learning a  $z$  vector and matching  $P(X)$  using maximum likelihood. The purpose of the VAE is to connect the relation between  $E_{Z \sim Q} P(X|z)$  and  $P(X)$  by a Gaussian kernel rather than learning an arbitrary distribution to optimize  $P(X)$ . This step fits the concept of adding noise while sampling.

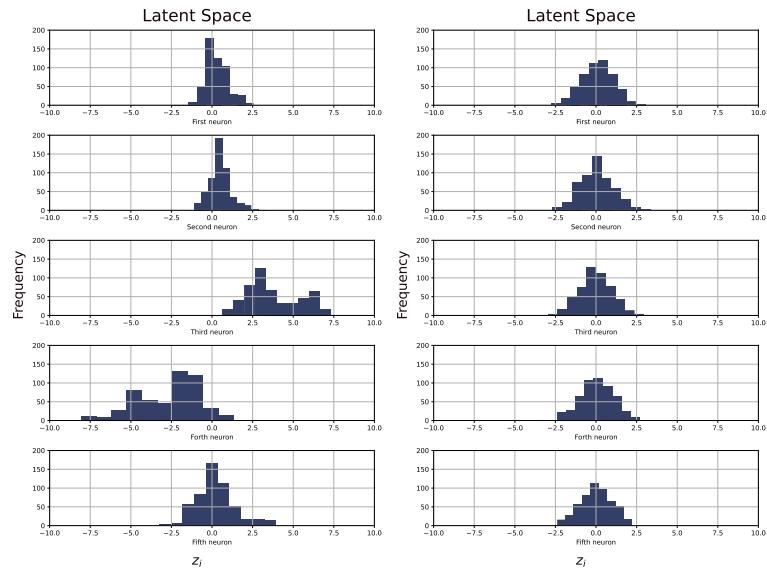


Figure 7. Comparing probability distribution of cAE and cVAE's latent space

Follow the Bayesian relation:

$$P(z|X) = \frac{P(X|z)p(z)}{P(X)}$$

$$P(z|X) = \frac{p(X|z)p(z)}{\int_z P(X|z)p(z)dz}$$

We have already defined  $p(z|X)$ , the problem here is to find the  $X$  by  $P(X|z)$ . VAE adopts the Kullback-Leibler divergence concepts between  $P(z|X)$  and  $Q(z)$  to solve the problem:

$$D[Q(z)||P(z|X)] = E_{z \sim Q}[\log Q(z) - \log P(z|X)]$$

$$D[Q(z)||P(z|X)] = E_{z \sim Q}[\log Q(z) - \log P(X|z) - \log P(z)] + \log P(X)$$

$$\log P(X) - D[Q(z|X)||P(z|X)] = E_{z \sim Q}[\log P(X|z)] - D[Q(z|X)||P(z)]$$

The equation above is core concept of VAE which indicates Q serves as a decoder and P serves as an encoder. At the left side of the equation shows the quantity  $P(X)$  which the models want to maximize (which fits the concept of maximum likelihood we mentioned before) and error term. At the right side of equation, we expect to maximize the expectation of the first term and minimize the KL divergence of the second term so the learned latent distribution closely approximates Normal.

### 3.5 Wasserstein Generative Adversarial Network (wGAN)

architecture<sup>1,3,11,13,15,18,21</sup>

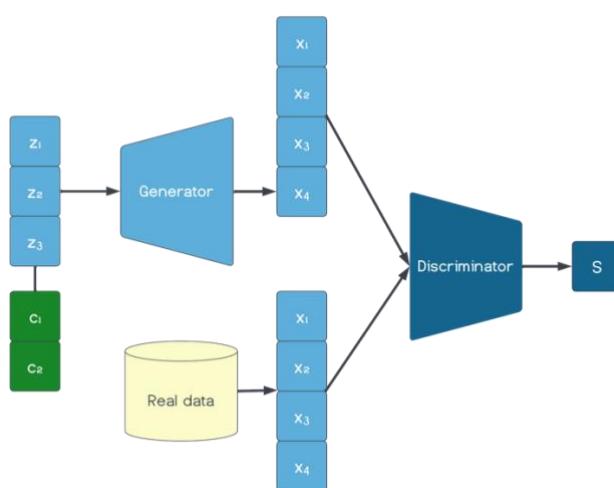


Figure 8. Schematic of the WcGAN model architecture.

The Generative Adversarial Network (GAN) is another generative approach which has been proven effective for learning complex distributions, most commonly for generating novel images. GAN is separated with two parts: generator and discriminator (two networks), which act as adversaries to each other during training.

The generator network is used to draw new samples  $x = g(z; \theta^g)$  from the unknown target distribution while the discriminator network attempts to classify between samples from the generator and samples from the training data by  $d(x; \theta^d)$ .

During training, the goal of the generator is to minimize the divergence between  $P_{\text{data}}$  and  $P_{\text{generator}}$  ( $g^* = \arg \min \text{Div}(P_{\text{generator}}, P_{\text{data}})$ ) while the discriminator tries to maximize the objective function ( $d^* = \arg \max V(g, d)$ ).

The objective function<sup>15</sup> is:

$$\min \max V(g, d) = E_{y \sim P_{\text{data}}} [\log(d(y))] + E_{y \sim P_{\text{generator}}} [\log(1 - d(y))]$$

At the right of the objective function, the left term represents the expected value of the training data which is expected to be maximized and the right term represents the expected value of generated data which is expected to be minimized. Ideally, when  $P_{\text{data}}$  and  $P_{\text{generator}}$  are converged, the discriminator will output  $\frac{1}{2}$  which means the generated data is indistinguishable from training data. In this formulation, the

generator tends to increase potential that the discriminator infers  $x = g(z; \theta^{(g)})$  with high possibility.

In practice, GAN is infamous for being difficult to train, since  $P_{\text{data}}$  and  $P_{\text{generator}}$  is hard to converge perfectly; it is common to cause underfitting. To solve the open problem, multiple algorithms have been developed for different circumstances. We adopt one of the most common algorithms and apply the Wasserstein distance to calculate the divergence of generated data and training data rather than rely on a binary classification from the discriminator network.

Wasserstein distance<sup>21,3</sup> has three distinct characteristics compared to other algorithms:

- 1) Measures the distance between discrete and continuous distributions.
- 2) Provides a scheme of how to transform one distribution into another.
- 3) Continuously transforms one distribution into another distribution, while maintaining the geometric characteristics of the distribution itself

Reference source not found., Error! Reference source not found., Error! Reference source not found., Error! Reference source not found.

## 4 Results and discussion

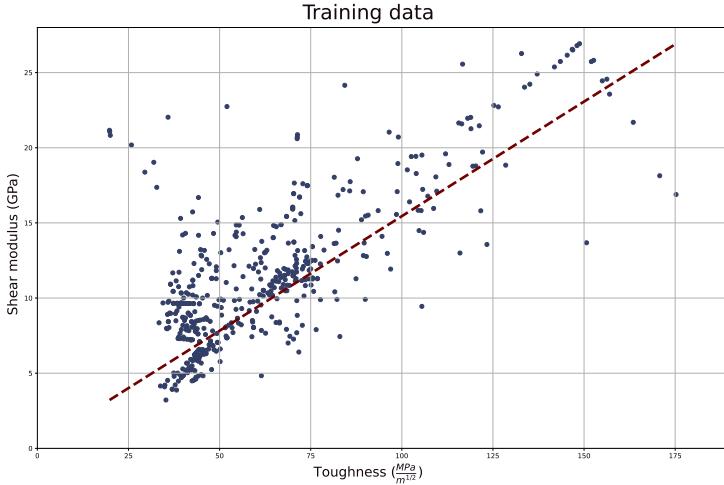


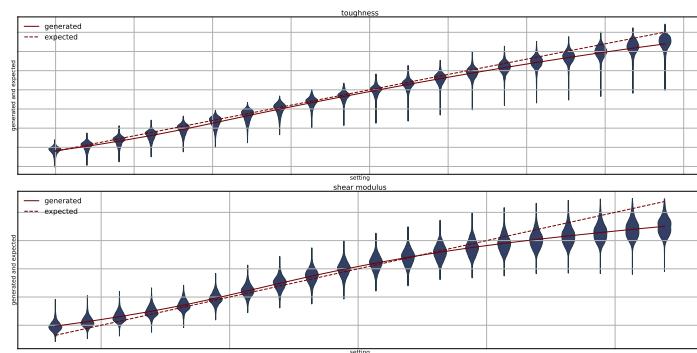
Figure 9. Training data are shown in blue symbols. We select 20 sets conditions along the red line as condition vectors for 3 model architectures to generate 2000 compositions respectively.

#### 4.1 Demonstration of inverse design

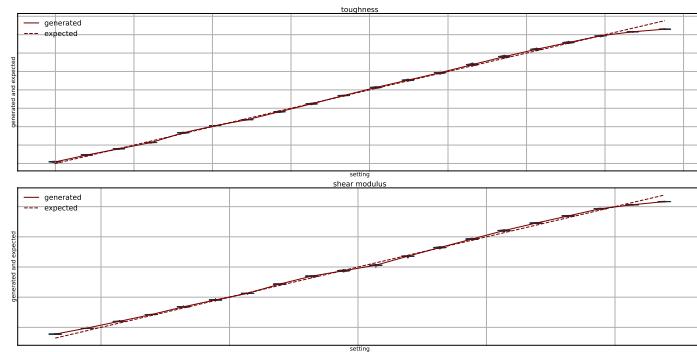
We select the 20 condition vectors varying together (i.e., along the red line shown in Fig. 9) to generate the results by the three different generative model architectures. The results of the inverse models are evaluated by surrogate forward model and shown in Fig. 10. We also select 25 condition vectors varying independently over the training domain shown in Fig. 9 and generate 2000 compositions at each point. The results are evaluated by the same forward surrogate model and shown in Fig. 11 (red dots represent conditions while blue color intensity represents probability density of finding a point in that pixel).

Broadly speaking, each of the three models can generate compositions whose value is biased toward the desired conditions (mechanical properties). However, it is clear that the models perform distinctly under different conditions. For example, at the

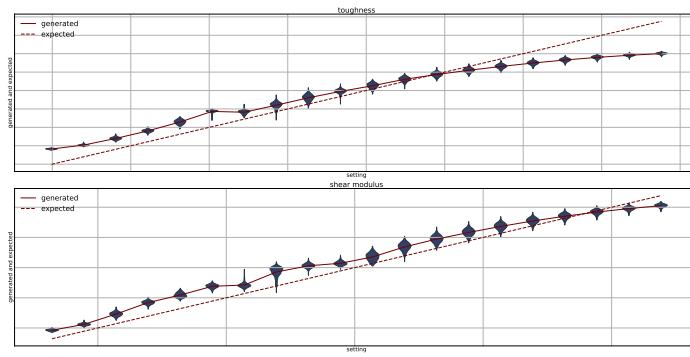
bottom right of each panel in Fig.11, the three models struggle to generate materials with accurate shear modulus and toughness value compared to when the target condition is located in the center of the training domain. This happens because there are few training samples whose value was close to the desired condition in training data (refer to Fig. 9). The more datapoints near the desired conditions, the more accurate the models will be in generating new samples since this was weighted more strongly in the training procedure.



(A)

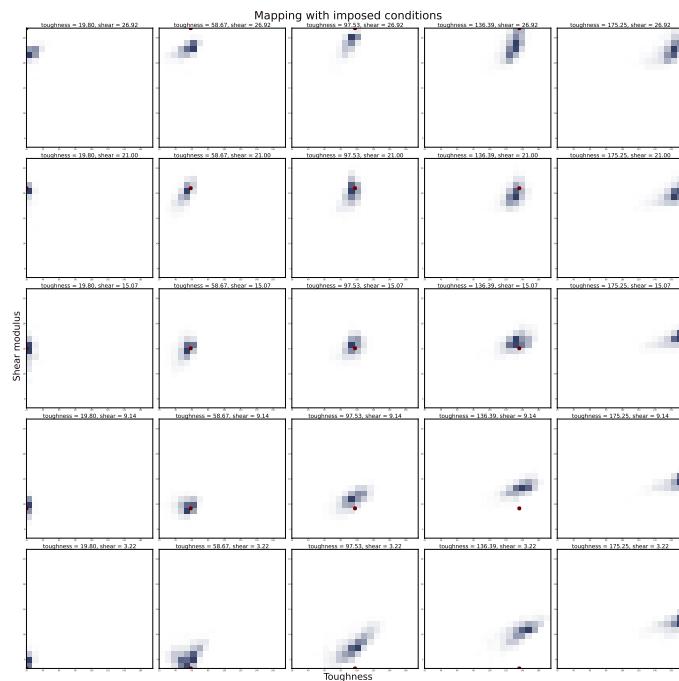


(B)

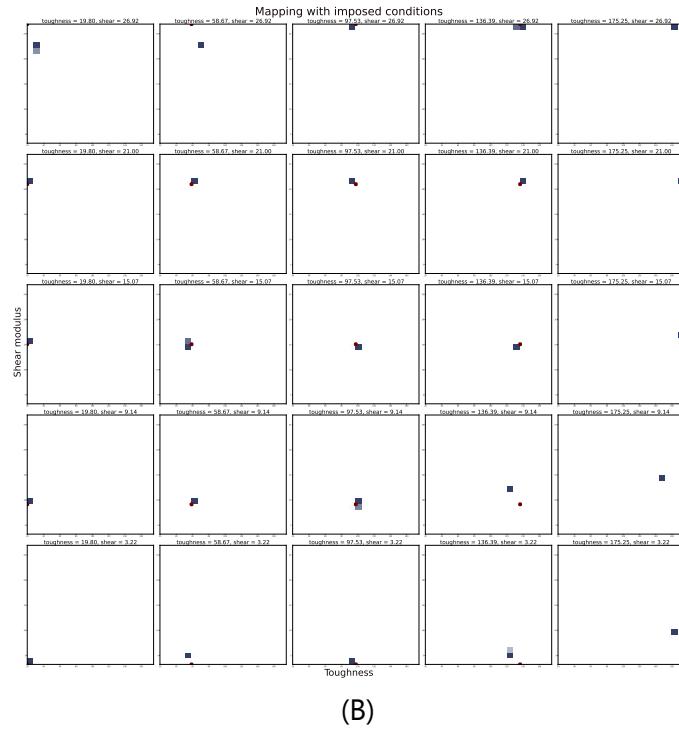


(C)

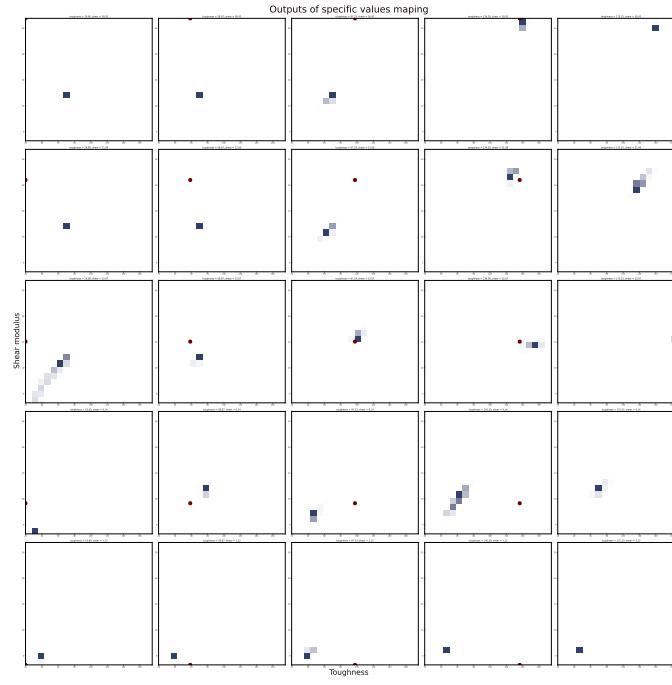
Figure 10. Parity plots for samples generated by (A) cAE, (B) cVAE, and (C) WcGAN. Each distribution shows the accuracy and precision from generating 2000 samples with a different condition along the dotted line which is also shown in Fig 9. The upper diagram shows the toughness results and the other show the shear modulus results. The solid line connects the average value of each distribution.



(A)



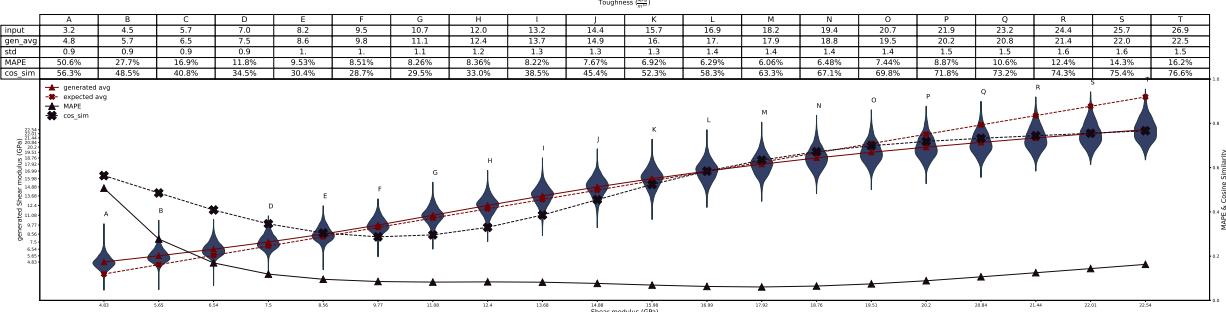
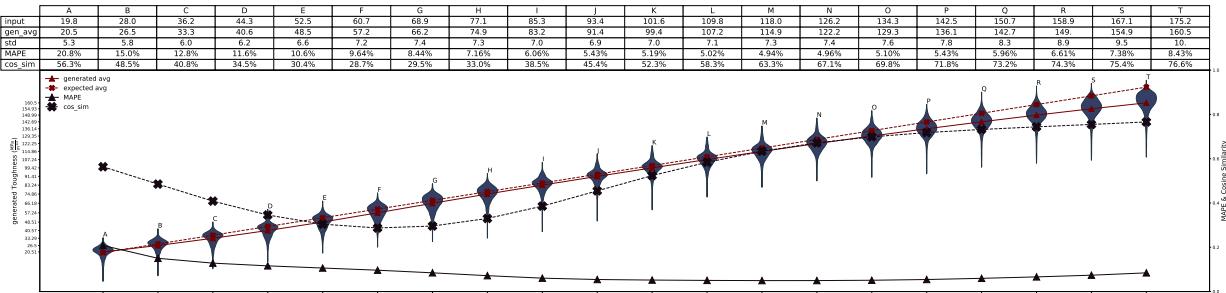
(B)



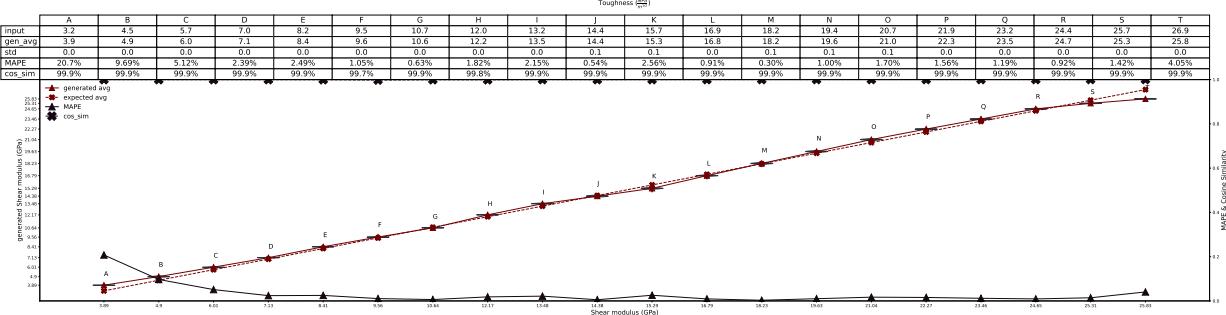
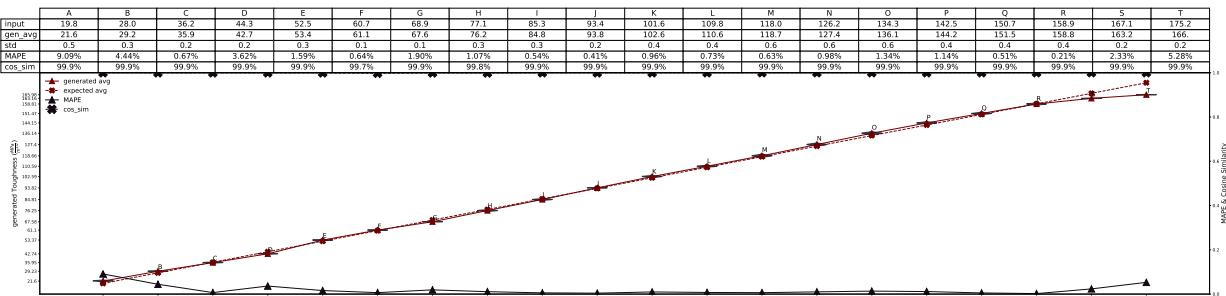
(C)

Figure 11 Empirical 2D probability distributions from (A) cAE, (B) cVAE, and (C) WcGAN. Each subplot represents the samples drawn for a single imposed condition vector, where red dots represent the target condition. Darker blue represents more samples located in the pixel (white indicates zero samples observed).

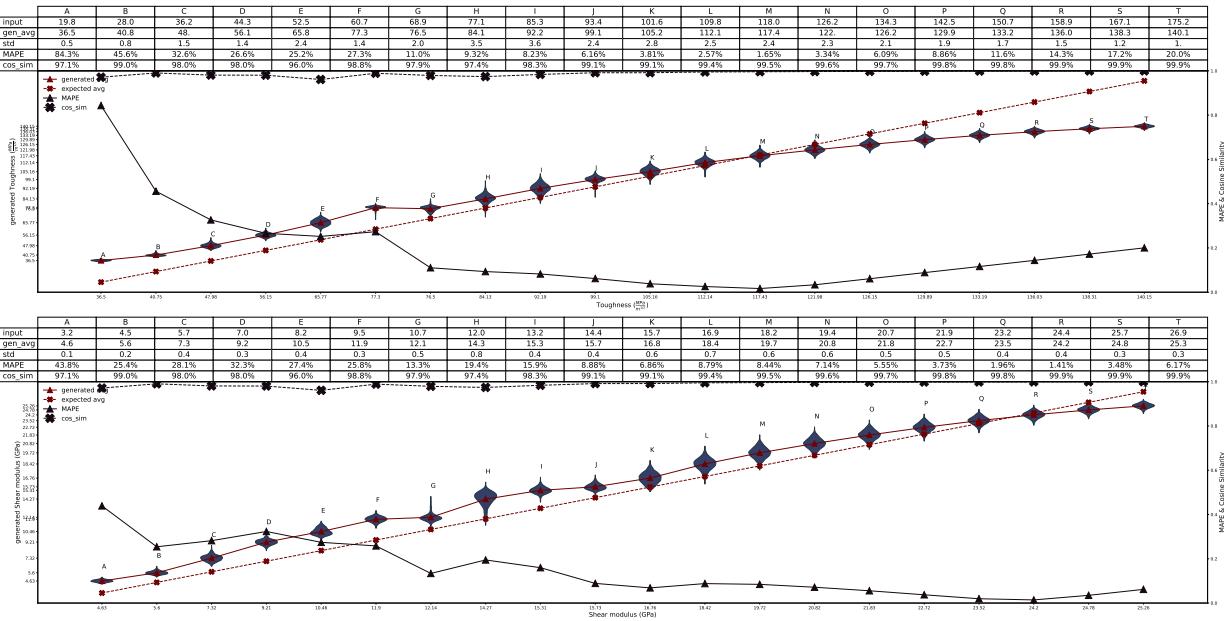
## 4.2 Quantitative comparison of model designs



(A)



(B)



(C)

Figure 12. Cosine similarity and MAPE by (A) cAE (B) cVAE (C) WcGAN

	cAE	cVAE	WcGAN
MAPE	10.51%	2.51%	16.51%
Cosine Similarity	53.44%	99.95%	98.87%

Table 1. Statistics of average MAPE and Cosine Similarity of three models.

We choose MAPE and Cosine similarity index to quantify the models with the generated compositions. MAPE means absolute percentage error, which is defined as:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

In our case, it indicates how accurately the generated materials' properties (shear modulus and toughness) fitted the target condition vector. Cosine similarity indicates how similar the generated compositions are; a lower value indicates greater

diversity among the generated compositions, which is desirable to avoid mode collapse where all the generated samples are the same.

From the MAPE curve in three models, it seems MAPE relates to the sparsity of datapoints around condition vectors: with more datapoints around, the MAPE value decreases. At the beginning and the end of the curve, MAPE is higher than the middle, and the datapoints are sparser than the middle.

For the Cosine similarity statics, the generated compositions from WcGAN and cVAE are highly similar to each other under the same conditions which implies the two models have experienced the mode collapse issue. This usually happens because of a small training dataset. The cosine similarity of compositions generated by cAE is high at the beginning stage but decreases until toughness is about  $70 \text{ MPa}/\text{m}^{1/2}$  or shear modulus is about 11 GPa, then increases again until the end. This also related to the sparsity of the datapoints around the applied condition vector, where the denser the datapoints near condition properties, the lower the cosine similarity. This means the model can generate more diversified compositions near conditions with more examples in the training dataset.

In short, we can conclude that with more datapoints around the condition property, our generated compositions' properties fit better to the input condition properties and are also more diverse (thus the model is more likely to give novel materials here).

### 4.3 Sample compositions comparison

From the random samples results of compositions by three generative models (Fig. 12 to 14) with shear modulus at 5 GPa and 21 GPa, we observe certain trends. Lower shear modulus elements are more prominent when the target shear modulus is low: Ti at 44 GPa, V at 47 GPa, and Nb at 38 GPa. Higher shear modulus elements are more prominent for a target of 21 GPa: W at 161 GPa, Re at 178 GPa, and Ru at 173 GPa. These trends fit the physics principles at play. It is clear to observe that the compositions generated by WcGAN and cVAE lack diversity which are related to mode collapse issue exposed by the cosine similarity metric; although they are not exactly the same, the differences are too small to distinguish each other visually.

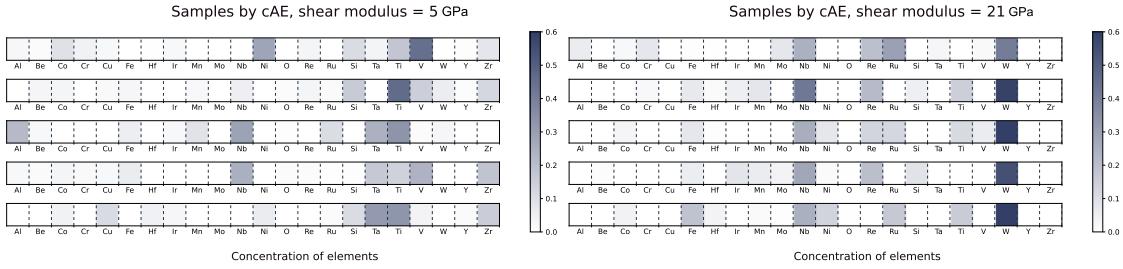


Figure 12. Generated samples by cAE. Left with 5 GPa shear modulus, right with 21 GPa shear modulus.

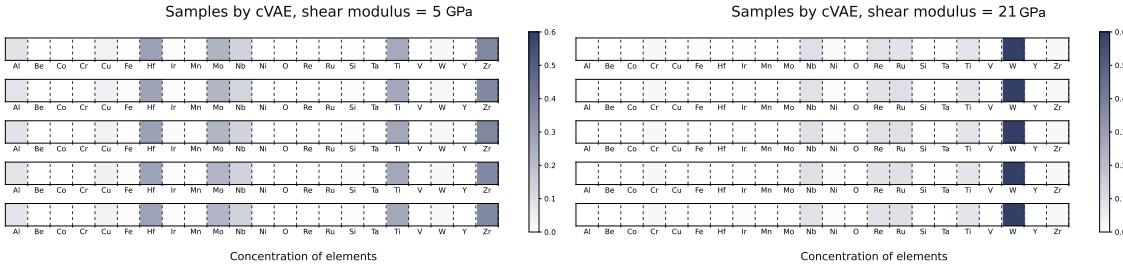


Figure 13. Generated samples by cVAE. Left with 5 GPa shear modulus, right with 21 GPa shear modulus

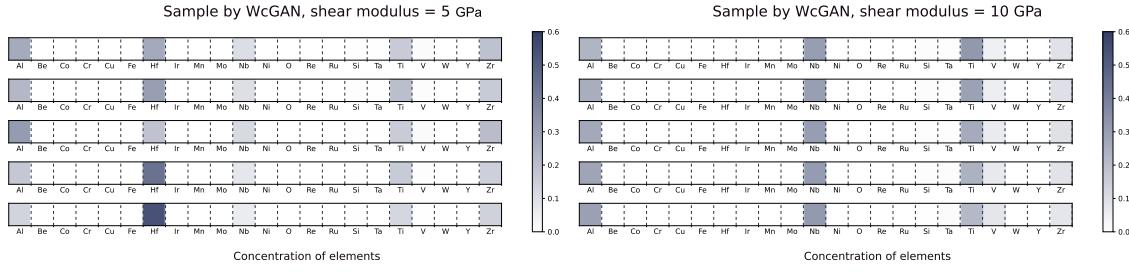


Figure 14. Generated samples by WcGAN. Left with 5 GPa shear modulus, right with 10 GPa shear modulus

#### 4.4 Latent space analysis

As theories indicated in Section 3, we can evaluate the structure of the latent space after model training by fixing the conditions and varying the latent code. The compositions generated by cVAE should have more gradual change from one point in the latent space to the other compared with cAE since cVAE has applied the kernel regularization trick (i.e., adding Gaussian noise to the samples).

Fig. 15 shows the sample results from cAE model under toughness with  $60 \text{ MPa/m}^{1/2}$  and shear modulus 15 GPa. The proportion of Ta and Mo elements is increasing, while Cr is decreasing from top to bottom material design. The cAE model still gives a continuous change in the result, but from the third to forth sample we also observe an abrupt decrease in the Nb composition compared to element Ta.

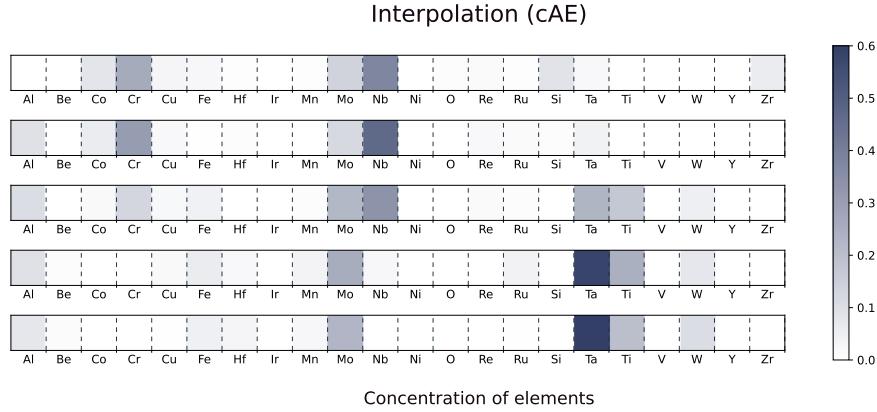


Figure 15. Compositions obtained by linearly interpolation from top to bottom with using cAE at fixed conditions

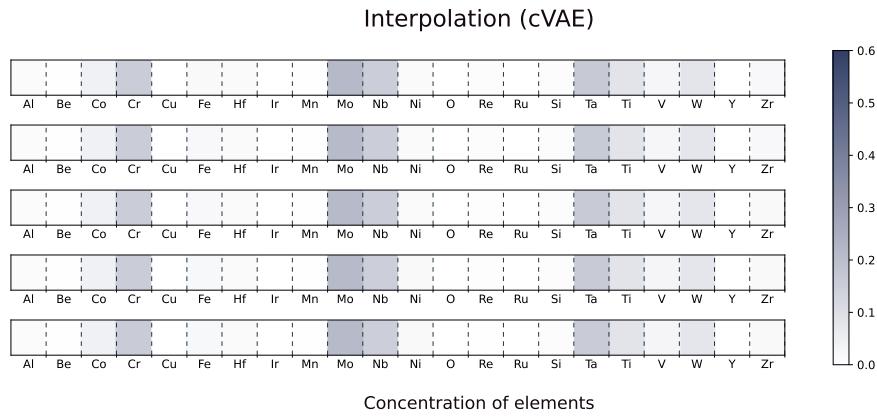


Figure 16. Compositions obtained by linearly interpolation from top to bottom with using cVAE at fixed conditions

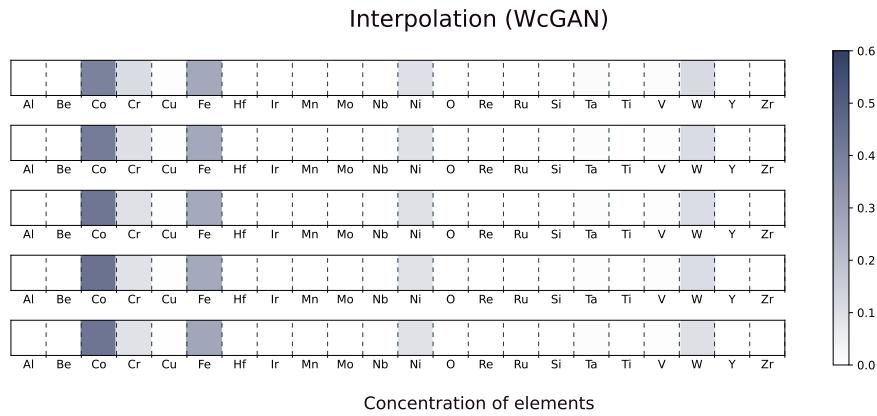


Figure 17. Compositions obtained by linearly interpolation from top to bottom with using WcGAN at fixed conditions

Figures 16 and 17 demonstrate the sample results of cVAE and WcGAN. We postulated that the problem arises from lack of training data, and as a result these

two models lack enough diversity to create multiple results. Therefore, we are not able to compare how the latent code variation would lead to material compositions. Following the results in the paper, Adversarial Mixup Resynthesizers<sup>[16]</sup>, we expect that after preprocessing, interpolation results will become smoother than cAE results (e.g., scenarios like the abrupt change in Nb would become rare).

## 4.5 Novelty of models

We randomly fetch 100 compositions from our dataset and use their properties as condition vectors to generate compositions with three models respectively, then calculate cosine similarity between generated samples and real composition to illustrate novelty value. Results are shown in Fig. 18 and Table 2.

Based on Table 2, cAE acquires lowest mean cosine similarity value; cVAE has similar performance with cAE either on distribution or mean value; WcGAN has possible of creating novel compositions corresponded to the minimum value observed in distribution plot.

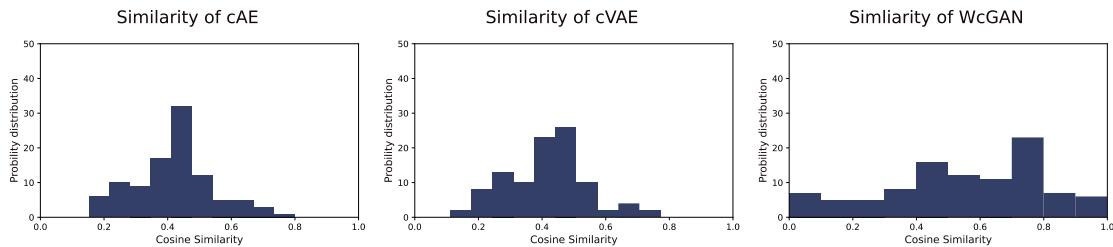


Figure 18. Cosine Similarity probability distribution of three models by comparing real composition and generated samples

	cAE	cVAE	WcGAN
Cosine Similarity	40.46%	41.36%	55.15%

Table 2 Mean Cosine Similarity of three models

## 5 Conclusion and future work

In agreement with theory, WcGAN and cVAE have greater accuracy -- as measured by MAPE -- than cAE. However, according to Section 4-2, these models have suffered from mode collapse and cannot create diversified samples. This is expected to be a result of the limited training data, and our priority in the future should become eliminating the effect of the small dataset. Unfortunately, acquiring new training data by experiments is the hardest part of the HEA design and the core challenge of materials research.

We also compared the samples of cVAE and WcGAN to find that WcGAN creates more novel compositions that are not found in our dataset under the same condition. In contrast, the compositions from cVAE are very similar to the compositions in the existing dataset which fits our assumption that autoencoder model gives out the results by linear combination within data points. In general, we conclude that based on the results above, cAE is able to create versatile compositions and keep at lower similarity; cVAE is capable of generating compositions with high accuracy; WcGAN has ability of creating novel compositions compared to training data.

At this time, using regression models is a great way to apply machine learning to HEA research. Several solutions, such as using simpler regression model or pre-trained models, have been found to be useful in mitigating the problem of small training datasets. In the future, we will put our effort towards finding other algorithms that can create novel compositions with limited data.

## 6 Bibliography

1. Aggarwal K, Kirchmeyer M, Yadav P, Keerthi SS, Gallinari P. (2019). Regression with Conditional GAN.
2. Arindam Debnath, Adam M. Krajewski, Hui Sun, Shuang Lin, Marcia Ahn, Wenjie Li, Shanshank Priya, Jogender Singh, Shunli Shang, Allison M. Beese, Zi-Kui Liu, Wesley F. Reinhart. (2021). Generative deep learning as a tool for inverse design of high entropy refractory alloys.
3. Arjovsky M, Chintala S, Bottou L. (2017). Wasserstein generative adversarial networks.
4. Benjamin Sanchez-Lengeling, Alán Aspuru-Guzik. (2018). *Inverse molecular design using machine learning: Generative models for matter engineering.*
5. Bhowmik A, Castelli IE, Garcia-lastra JM, Jørgensen PB, Winther O, Vegge T. (2019). *A perspective on inverse design of battery interphases using multi-scale modelling, experiments and generative deep learning.*
6. Chen J, Zhou X, Wang W, et al. (n.d.). *A review on fundamental of high entropy alloys with promising high-temperature properties.*
7. Cheng Wen, Y. Z. (2019). *Machine learning assisted design of high entropy alloys with desired property.*
8. Christopher Beckham, Sina Honari, Vikas Verma, Alex Lamb, Farnoosh Ghadiri, R Devon Hjelm, Yoshua Bengio, Christopher Pal. (2019). *On Adversarial Mixup Resynthesis.*

9. Chuan Zhang, Fan Zhang, Shuanglin Chen and Weisheng Cao. (2012). Computational Thermodynamics Aided High-Entropy Alloy Design.
10. Chun-Teh Chen, Grace X. Gu. (n.d.). *Generative deep neural networks for inverse materials design using backpropagation and active learning.*
11. Dan Y, Zhao Y, Li X, Li S, Hu M, Hu J. (2020). Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials.
12. Diederik P. Kingma, Max Welling. (2013). Auto-encoding variational Bayes.
13. Dong Y, Li D, Zhang C, et al. (2020). Inverse design of two-dimensional graphene/h-BN hybrids by a regressional and conditional GAN.
14. Dongbo Dai , Tao Xua , Xiao Wei , Guangtai Ding, Yan Xu, Jincang Zhang, Huiran Zhang. (2020). *Using machine learning and feature engineering to characterize limited material datasets of high-entropy alloys.*
15. Goodfellow I, Bengio Y, Courville A. (2016). *Deep learning.*
16. Jingjing Ruan, Weiwei Xu, Tao Yang, Jinxin Yu, Shuiyuan Yang, Junhua Luan, Toshihiro Omori, Cuiping Wang, Ryosuke Kainuma, Kiyohito Ishid, Chain Tsuan Liu, Xingjun Liu. (2020). *Accelerated design of novel W-free high-strength Co-base superalloys with extremely wide  $\gamma/\gamma'$  region by machine learning and CALPHAD methods.*
17. Kyle Mills, Kevin Ryczko, Iryna Luchak, Adam Domurad, Chris Beeler, Isaac Tamblyn. (2019). Extensive deep neural networks for transferring small scale learning to large scale systems.
18. Li K, Malik J. (2018). On the implicit assumptions of gans.
19. Li Q, C. W. (2017). *On sluggish diffusion in Fcc Al-Co-Cr-Fe-Ni high entropy alloys: an experimental and numerical study.*
20. Lim J, Ryu S, Kim JW, Kim WY. (2018). Molecular generative model based on conditional variational autoencoder for de novo molecular design.

21. Martin Arjovsky , Soumith Chintala , and Léon Bottou. (2017). Wasserstein GAN.
22. Melia MA, Whetten SR, Puckett R, et al. (2020). *High-throughput additive manufacturing and characterization of refractory high entropy alloys.*
23. Melia MA, Whetten SR, Puckett R, et al. (2020). High-throughput additive manufacturing and characterization of refractory high entropy alloys.
24. Ming-Huang Tsai, Jien-Wei Yeh. (2014). High-Entropy Alloys: A Critical Review.
25. Nan Qua, Yong Liua, Mingqing Liaoa, Zhonghong Laia, Fei Zhoua, Puchang Cuia, Tianyi Hana, Danni Yang, Jingchuan Zhu. (2019). *Ultra-high temperature ceramics melting temperature prediction via machine learning.*
26. Philips NR, Carl M, Cunningham NJ. (n.d.). *New opportunities in refractory alloys.*
27. Sanchez-Lengeling B, Aspuru-Guzik A. (2018). Inverse molecular design using machine learning: Generative models for matter engineering.
28. Sathiyamoorthi Praveen, Hyoung Seop Kim. (2017). High-Entropy Alloys: Potential Candidates for High-Temperature Applications – An Overview.
29. Schleder GR, Padilha ACM, Acosta CM, Costa M, Fazzio A. (2019). From DFT to machine learning: recent approaches to materials science-a review.
30. Schmidt J, Marques MRG, Botti S, Marques MAL. (2019). Recent advances and applications of machine learning in solid-state materials science.
31. Senkov ON, M. D. (2018). *Development and exploration of refractory high entropy alloys-A review.*
32. ShengGuo, Qiang Hu, Chun Ng, C.T.Liu. (2013). More than entropy in high-entropy alloys: Forming solid solutions or amorphous phase.
33. Siwar Chibani, François-Xavier Coudert. (2020). *Machine learning approaches for the prediction of materials properties.*

34. Wu Y, Si J, Lin D, et al. (2018). Phase stability and mechanical properties of AlHfNbTiZr high-entropy alloys.
35. Z, L. (2018). Ocean of Data: Integrating first-principles calculations and CALPHAD modeling with machine learning.
36. Huang W, Martin P, Zhuang HL(2019). Machine-learning phase prediction of high-entropy alloys
37. Li Y, Guo W(2019). Machine-learning model for predicting phase formations of high-entropy alloys
38. Kaufmann K, Maryanovsky D, Mellor WM, et al (2020). Discovery of high-entropy ceramics via machine learning