Deep Learning Enabled Inorganic Material Generator

by

Yashaswi Pathak, Karandeep Singh Juneja, Girish Varma, Masahiro Ehara, U Deva Priyakumar

Report No: IIIT/TR/2020/-1

Centre for Computational Natural Sciences and Bioinformatics
International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2020

Deep Learning Enabled Inorganic Material Generator

Yashaswi Pathak,[†,¶] Karandeep Singh Juneja,[†,¶] Girish Varma,[†] Masahiro Ehara,[‡] and U Deva Priyakumar[*,†]

[†] *International Institute of Information Technology, Hyderabad 500 032, India*

[‡] *Research Center for Computational Science, Institute for Molecular Science, Okazaki 444-8585, Japan*

[¶] *Authors contributed equally*

E-mail: deva@iiit.ac.in

Abstract

Recent years have witnessed utilization of modern machine learning approaches for predicting properties of material using available datasets. However, to identify potential candidates for material discovery, one has to systematically scan through a large chemical space and subsequently calculate the properties of all such samples. On the other hand, generative methods are capable of efficiently sampling the chemical space and can generate molecules/materials with desired properties. In this study, we report a deep learning based inorganic material generator (DING) framework consisting of a generator module and a predictor module. The generator module is developed based upon conditional variational autoencoders (CVAE) and the predictor module consists of three deep neural networks trained for predicting enthalpy of formation, volume per atom and energy per atom chosen to demonstrate the proposed method. The predictor and generator modules have been developed using a one hot key representation of the material composition. A series of tests were done to examine the robustness of the predictor models, to demonstrate the continuity of the latent material space, and its ability to generate materials exhibiting target property values. The DING architecture proposed in this paper can be extended to other properties based on which the chemical space can be efficiently explored for interesting materials/molecules.

Introduction

In the last few years, deep learning methods have had profound success in several applied areas of computer science such as computer vision, natural language processing and robotics.[1] This has led to application of these algorithms to tackle problems across different backgrounds including in natural sciences.[2–6] Recently, there has been a surge in incorporating data driven approaches in several aspects of molecular/material design. Tremendous progress has been made in molecular property prediction (for eg. quantum mechanical energies), conceiving of retrosynthetic pathways, drug design, protein structure prediction, where machine learning has played a crucial role.[7–11] Availability of large datasets in general has increased the relevance of data driven machine learning based approaches in the area of molecular sciences.[12–18]

In the past decade, machine learning techniques have been widely used by researchers to address many challenges in the field of material science and engineering.[2,19–35] Ward et al.[27] proposed a set of 145 hand engineered features based on stoichiometric attributes, elemental property statistics, electronic structure attributes, ionic compound attributes which can be used for a broad range of datasets. Jha et al. in their work[20] introduced a deep learning framework called ElemNet to predict formation enthalpy of materials directly from its elemental composition, eliminating the need of domain knowledge to hand engineer features. Such data driven approaches have been very successful[2,26,28,36] and provide a faster replacement for the time taking density functional theory[37] calculations to predict properties of materials.[38] However, these conventional methods that enable accurate predictions of material properties rely upon efficiently scanning the material space[20,22] in a high-throughput manner to identify potential candidates as useful materials (see 'conventional approach' in Figure 1). This procedure also biases the way in which generation of materials is done during the screening protocol. For example, a systematic sampling of the material space might end up with only materials that belong to a predefined pattern, like a binary compound of type AB or a ternary compound of type A_2B, etc. Due to the vast and discrete nature of the material space, the success rate of such combinatorial sampling is low.

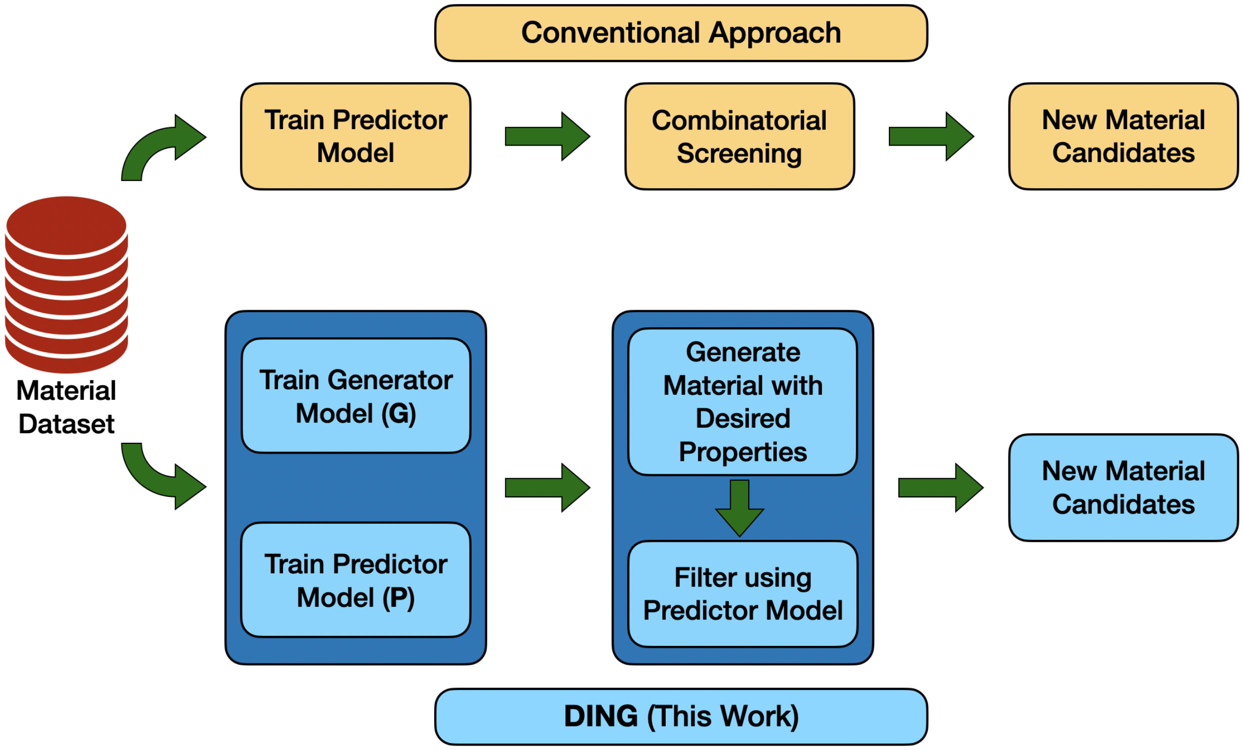


Figure 1: Schematic representation of the the Deep INorganic materials Generator (DING) method proposed in this study as compared to conventional approaches of generating potential candidates for material discovery. In conventional approach, first a predictor model is trained, then materials are generated combinatorially and the potential candidates are screened using the predictor network. In DING model, generator and predictor models are trained and then the generator model is used to identify potential candidates with desired properties, which are finally evaluated using the predictor network. This results in a small search space and achieves efficient sampling of the material space.

Generative models based on deep learning and reinforcement learning methods have shown great promise in the paradigm of *de novo* design of small organic molecules.[8,39–42] Gómez-Bombarelli et al.[39] showed that a model based on variational autoencoders[43] can be used to describe a molecule in a latent space where they are expressed as real vectors. On this latent space, because of its continuous and differentiable nature, a gradient based optimization can be used to optimize properties of molecule and hence identify molecules with desired properties directly. Guimaraes et al.[40] proposed Objective-Reinforced Generative Adversarial Networks (ORGAN) to generate small organic molecules with target properties whereas Popova et al.[8] and You et al.[42] used deep reinforcement learning methods to achieve similar goals. While reasonable progress has been made in the use of generative ML methods for inverse design of drug–like molecules, efforts towards inverse material design are comparatively fewer. Noh et al.[44] proposed iMatGen, an image-based materials generator using a combination of autoencoders and variational autoencoders forV_xO_y system. Hoffmann et al[45] utilized the 3-D positions of atoms in a molecule in an encoder-decoder framework to generate materials. Recently, Nouira et al.[46] utilized Generative adversarial networks in their work CrystalGAN to produce stable ternary structures. Such applications of generative models to directly design inorganic materials with target properties are expected to contribute further to the general field of material design.

Attempts towards *de novo* molecular generation have primarily used four different approaches, namely, genetic algorithms, variational autoencoders (VAE), reinforcement learning (RL) and generative adversarial networks (GAN).[40,47–51] Generative models such as VAE attempts to learn a distribution that is close to the unknown probability distribution of the given data. With rapid advances in deep learning, neural networks have emerged as powerful approximators for these probability distribution functions. Deep neural network based generative models take latent vector as input, which is randomly sampled from a known distribution and produces data similar to the training data. There are many variants of deep generative models and in this work we have used an extension of VAE, namely the conditional variational autoencoder (CVAE).[52,53]

CVAE has shown remarkable performance as a generative model in the field of computer vision[54,55] and natural language processing.[56,57] Lim et al.[58] recently used a CVAE based model for de novo organic molecular design. Models based on CVAEs provide a continuous latent space along with control over properties which can be manipulated as desired during decoding process to generate molecules with target properties (for example, Lim et al. used logP values, molecular weight, number of hydrogen bond donors/acceptors and topological polar surface area as different properties). In this study, we report a deep learning model based on CVAE to generate inorganic material compositions that will show desired properties of energy per atom, volume per atom and formation enthalpy. Open quantum materials database (OQMD) database developed by Wolverton and coworkers[12,13] was used to train and develop the ML method. This method named Deep INorganic material Generator (DING) allows generation of inorganic material candidates by optimizing the property space as mentioned above (see Figure 1). For estimating the properties of the generated material candidates, we use separately trained prediction models, that given a material composition, is able to predict properties of interest with reasonable accuracy. We also design an appropriate feature vector which is more suitable for this task. Further, we perform various experiments to show the continuity of the generated latent space, ability of the model to generate materials and the robustness of DING.

Theory and Methods

The framework proposed in this study, DING, (see figure 1) consists of a generator module based on CVAE[52] and a predictor module. Candidate materials are proposed by the generator module and the candidates are then filtered using the predictor module. This section gives the details of the dataset, feature vector used, methods used and the training procedure used. The code for this work along with examples is available at: `https://www.github.com/devalab/DING`

Dataset

The open quantum materials database (OQMD) is used to train and test the generator and predictor modules proposed in the DING framework.[12] OQMD is a high-throughput database, which currently consists of nearly 637,644 density functional theory (DFT) calculations of compounds from the Inorganic Crystal Structure Database (ICSD).[59] In this study, only unique composition of materials were taken and three properties, namely formation enthalpy, volume per atom and energy per atom were used to test the method. Formation enthalpy is defined as the energy of forming a compound from its constituent elements. For materials with more than one structure for a given composition, the one with the lowest formation enthalpy was used. This is essential since the feature vector used for this method encodes the composition not the three dimensional structure (see later). Additionally, only those composition for which the values for the three properties are available were retained. Compounds that had atomicity of more than 10 for a single element were also removed from the dataset. This eliminated very few entries in the dataset, and was done to limit the input dimensions. After these prepossessing steps, 272,064 data points was obtained. Figure 2 shows the distribution of formation enthalpy, energy per atom and volume per atom. From this dataset, randomly chosen 10% was used as the test set and the rest 90% was used for training and validation by doing a 80:20 random split.

Material Descriptor

Ward et al.[27] introduced a general set of 145 features derived from a material's stoichiometric attributes, elemental property statistics, electronic structure attributes and ionic compound attributes that can be used for multiple property prediction. This way, information related to materials were explicitly given in the feature representation. Jha et al.[20] in their work ElemNet used a feature vector made up only from elemental composition. This contained a bin corresponding to each element in the dataset and the fractional value of that element in the material was put in each of the bins. This descriptor gives the model complete

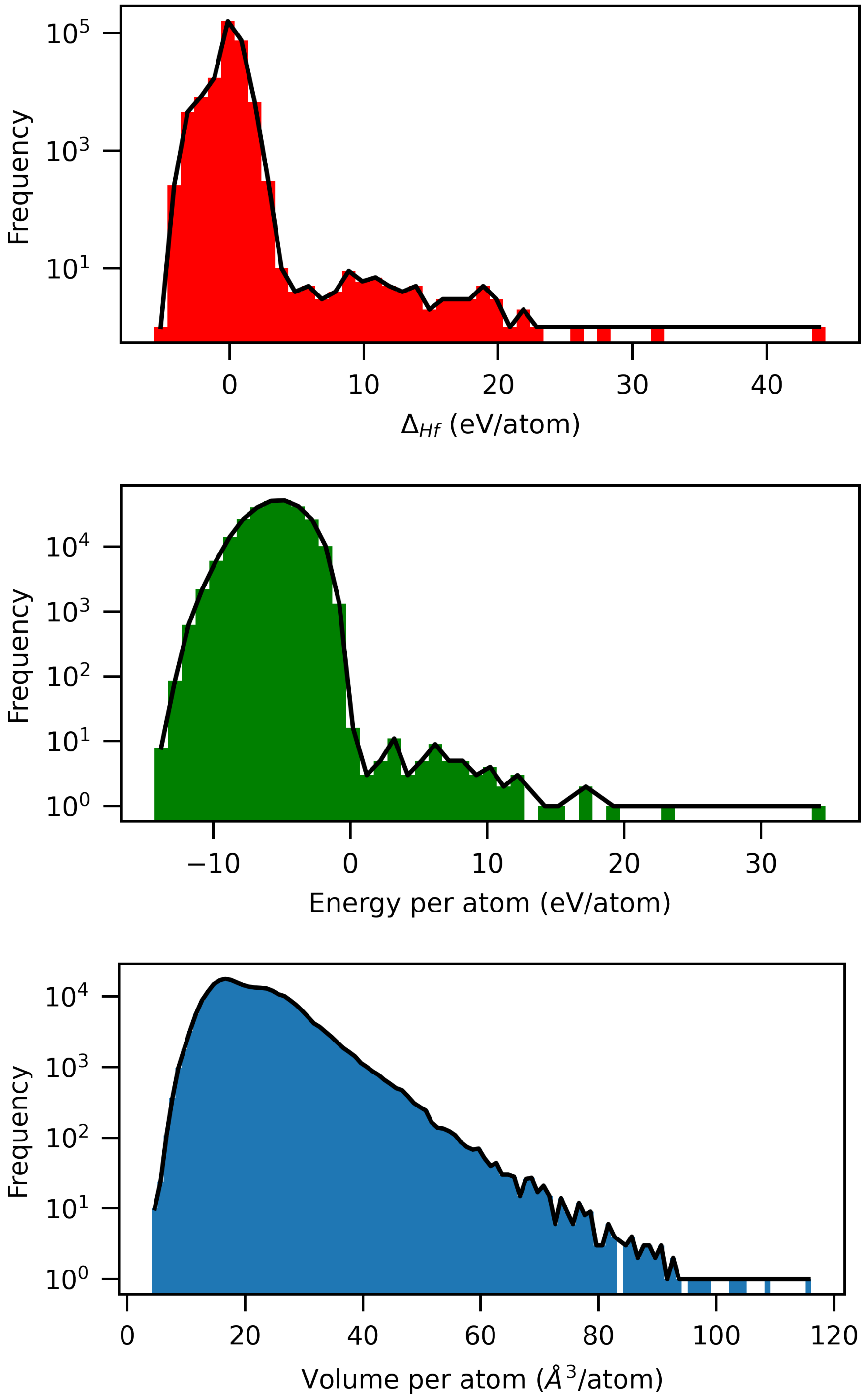
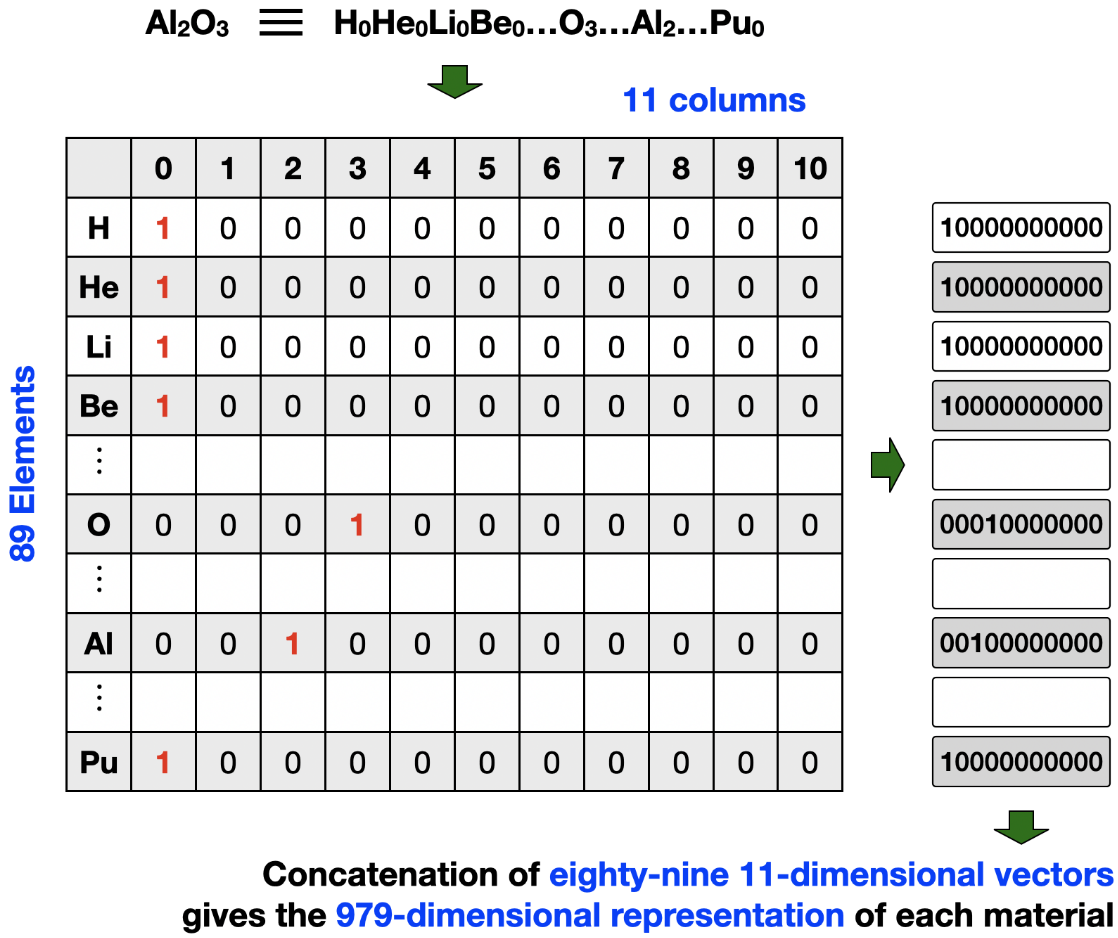


Figure 2: Distribution of formation enthalpy, energy per atom and volume per atom in the final dataset.

independence to self learn the desired chemistry whereas in the former one, domain knowledge is supplied to the model explicitly. In this study, a feature vector constructed only from the elemental composition of the material is used. Each element is represented by a constant length vector (11 dimensions) and the total number of elements considered for the feature vector representation is 89. The one hot vectors corresponding to each element are then concatenated together to form a 979 dimensional vector representing each material in the dataset. Figure 3 represents the feature vector used here by taking Al_2O_3 as an example.



$Al_2O_3 \equiv H_0He_0Li_0Be_0 \ldots O_3 \ldots Al_2 \ldots Pu_0$

11 columns

89 Elements

	0	1	2	3	4	5	6	7	8	9	10
H	1	0	0	0	0	0	0	0	0	0	0
He	1	0	0	0	0	0	0	0	0	0	0
Li	1	0	0	0	0	0	0	0	0	0	0
Be	1	0	0	0	0	0	0	0	0	0	0
⋮											
O	0	0	0	1	0	0	0	0	0	0	0
⋮											
Al	0	0	1	0	0	0	0	0	0	0	0
⋮											
Pu	1	0	0	0	0	0	0	0	0	0	0

Concatenation of eighty-nine 11-dimensional vectors gives the 979-dimensional representation of each material

Figure 3: Feature vector used to represent the composition of the inorganic materials. Concatenation of all the one hot vectors (different rows) of elements in the dataset to represent the complete material. Al_2O_3 is used as an example to define the feature vector.

Predictor Module

In the DING framework, the predictor module is used to filter the candidates proposed using the generator model. The predictor module consists of three separate models, one each for formation enthalpy, energy per atom and volume per atom trained on the OQMD dataset. The feature vector introduced in the above section is used as the input. Each predictor model is an 8–layered deep neural network with dropout[60] layers at three layers (see Table 1). ReLU was used as the activation function for the hidden layers.

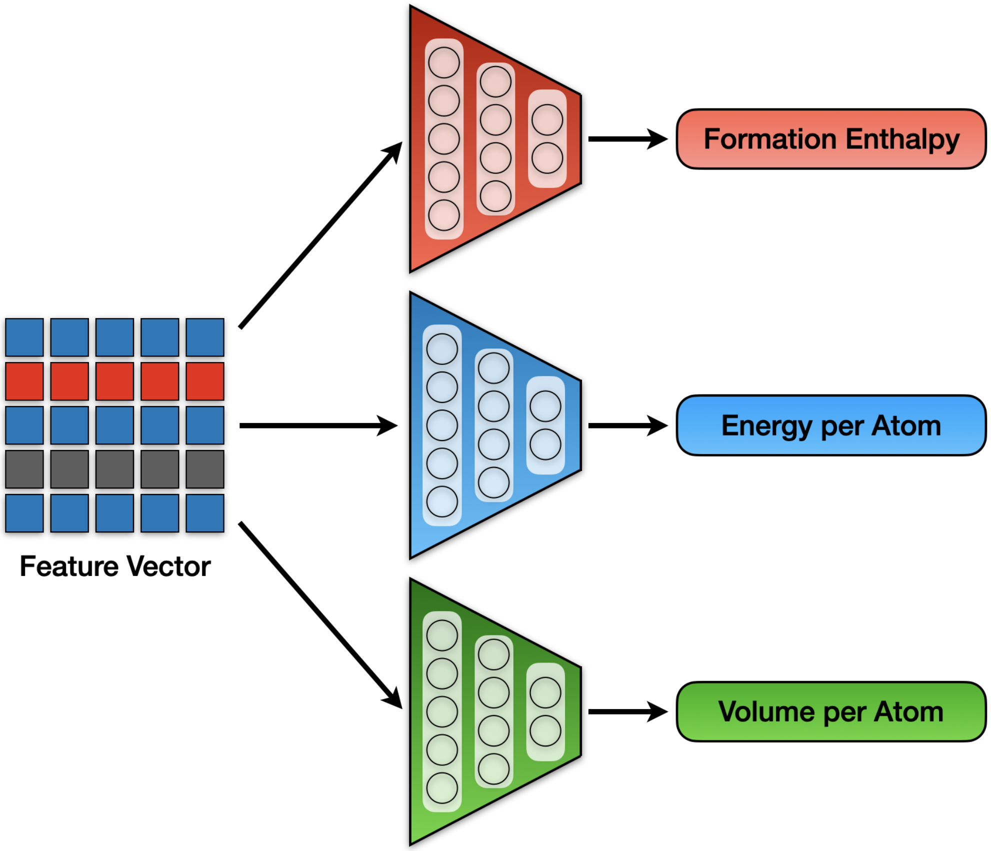


Figure 4: Schematic representation of the predictor module that consists of three deep neural networks predicting each of them predicting formation enthalpy, energy per atom and volume per atom. The architecture of these networks are given in Table 1.

Generator Module

Variational Autoencoders (VAE)

In an autoencoder (AE), we have a pair of deep neural networks, namely an encoder and a decoder. The encoder network converts the input feature vector to a fixed-dimensional vector, and the decoder network converts this fixed-dimensional vector back to the original

Table 1: Architecture of the deep neural networks in the predictor module.

Layer types	No of Units	Activation	Layer Positions
Dense	1024	ReLU	1^{st} to 2^{nd}
Dropout(0.7)	1024	-	After 2^{nd}
Dense	1024	ReLU	3^{rd}
Dropout(0.7)	1024	-	After 3^{rd}
Dense	512	ReLU	4^{th}
Dropout(0.5)	1024	-	After 4^{th}
Dense	128	ReLU	5^{th}
Dense	64	ReLU	6^{th}
Dense	32	ReLU	7^{th}
Dense	1	Linear	8^{th}

input feature vector. The goal of an autoencoder is to learn an identity function. The fixed-dimensional vector is generally referred as the *latent vector* z. The latent vector z acts as an information bottleneck in an AE, i.e. it is designed to learn the most statistically salient information in the data.[39] Autoencoders are designed as a dimensionality reduction framework, where the objective is to minimize the error in reconstructing the original input vector. There are no constraints on the latent vector z which may enable an autoencoder to act as a generative model. This constraint is incorporated in a VAE.

VAEs are one of the most popular and effective generative models. In VAEs, the latent vectors z are sampled from a normal distribution $\mathcal{N}(0, I)$, where I is the identity matrix. The objective function that needs to be optimized for VAEs, with input data X and latent vector z can be formulated as:

$$\mathbb{E}\left(\log P_{\Theta}(X \mid z)\right) - D_{KL}(Q_{\Theta'}(z \mid X) \parallel N(z)) \tag{1}$$

where $\mathbb{E}$ is the expectation value, P and Q are probability distributions and D_{KL} is the Kullback-Leibler divergence, which measures how different one distribution is from the other. The probability distributions $P_{\Theta}(X \mid z)$ and $Q_{\Theta'}(z \mid X)$ are learned by deep neural networks called the *decoder* and *encoder* respectively with learnable parameters Θ, Θ'. The first term is simply the reconstruction error for the datapoint X, while the second term measures the similarity between the probability distribution of the latent space *encoded* from the input data and the target probability distribution, $N(z)$, which is $\mathcal{N}(0, I)$.

Conditional Variational Autoencoders (CVAE)

In case of VAEs, it is not possible to control the properties of the data generated. The generation could be a random instance from the training distribution, the properties of which may vary widely. The objective is to generate materials with specific properties. Also, since the latent vector is sampled from a Gaussian, the distribution is unimodal. In such problems, simple machine learning models like regressors can generate data that is an average of all possibilities. CVAEs are VAEs that can learn multimodal probability distributions by adding a condition vector as an additional input during the generation process. The objective function of the CVAE with condition vector, c, (given as an input to both the encoder and the decoder) is given by,

$$\mathbb{E}\left(\log P_{\Theta}(X \mid z, c)\right) - D_{KL}(Q_{\Theta'}(z \mid X, c) \parallel N(z)) \tag{2}$$

Gómez-Bombarelli et al.[39] proposed a VAE molecule generator jointly trained with another network that predicts the properties of molecules. Then, gradient descent optimization was performed in the latent space to find the materials with desired properties. In the DING framework, the properties of the material that is to be generated are represented as the condition vector which is directly passed as an input to the CVAE. Thus, after the model is trained, it can generate new materials with properties close to the desired ones. A key difference between this and the approach taken by Gómez-Bombarelli et al. is that here, we do not need to perform optimization for every vector of properties after the generation.

The generator model in this study is a deep CVAE consisting of 7 hidden layers and as mentioned earlier, it can be described by three different pieces: the encoder, latent space and

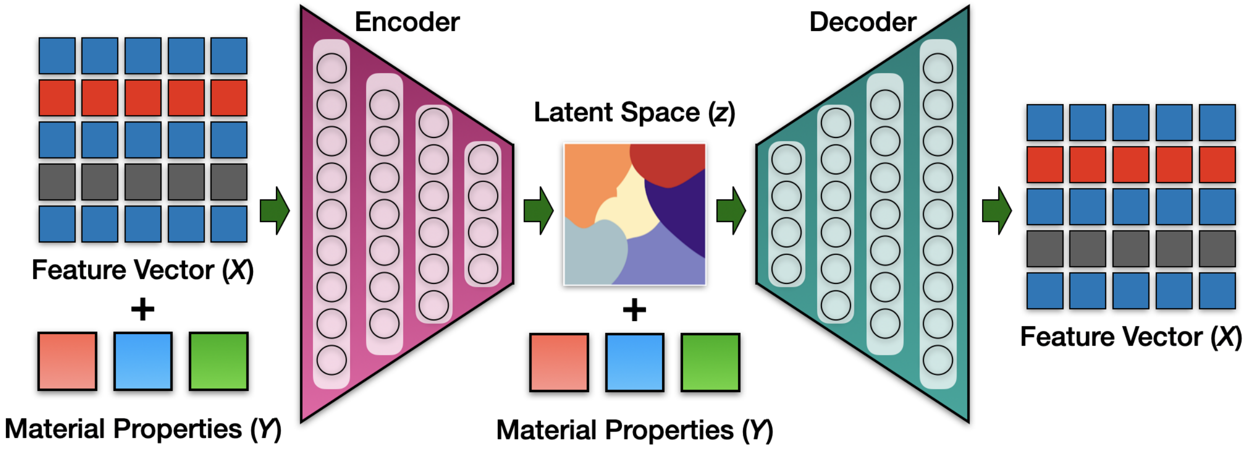


Figure 5: CVAE model used for generation of materials. The 979-dimensional one hot feature vector is concatenated with the respective properties of the material and is fed as input to the encoder network. The encoder networks encodes this information into a vector in the latent space (z). The decoder networks takes the property of the material along with its latent vector and regenerates the material.

the decoder (Figure 5). As mentioned above, the material compositions are represented by a 979-dimensional vector. During training, along with the feature vector X describing the compound, the three properties Y are also appended and hence, the dimension of the first layer becomes 982. The encoder has 3 hidden layers and its architecture is given in Table 2. The latent space Z has a dimension of 32. During decoding,the three properties Y of the material are appended again so that the decoder can be controlled later to manipulate these properties and reach the desired target material. The architecture of decoder is given in Table 2. Each layer in the model, except the last layer of decoder has a ReLU activation unit, whereas the last layer of decoder has a softmax activation function.

Table 2: The architecture of the encoder, latent space and the decoder network of DING model.

Network	Dimensions
Encoder	982:512:256:128:32
Latent Space	32
Decoder	35:128:256:512:982

Generation of materials

After the CVAE is trained, the decoder part of the model is used for generating materials. A random z is sampled out from $N(0, I)$, and is concatenated with the desired property values, which is then given as an input to the decoder model. The decoder model then outputs the composition of the material with the desired properties (See Figure 6). The predictor models then are used to predict the values of the three properties for further filtering/validation.

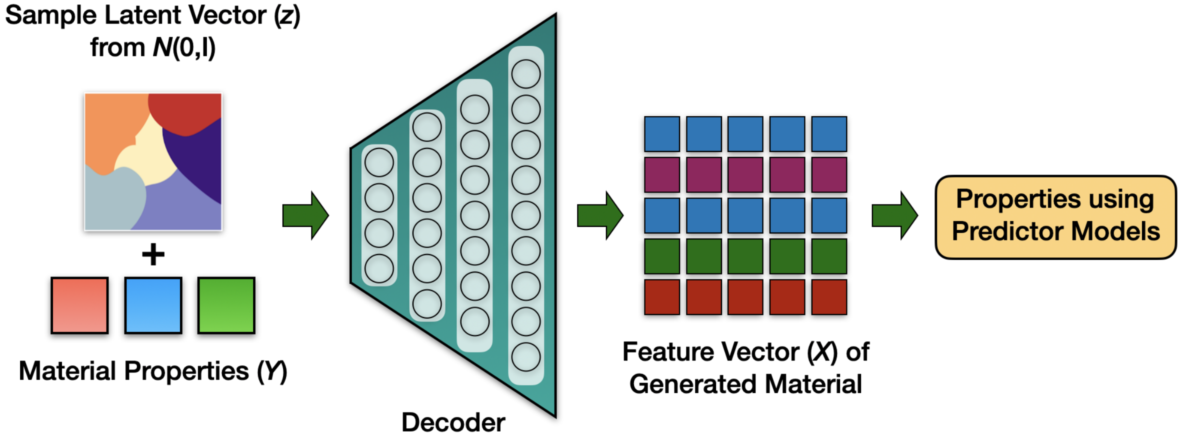

Sample Latent Vector (*z*) from *N*(0,I)
+
Material Properties (*Y*)
Decoder
Feature Vector (*X*) of Generated Material
Properties using Predictor Models

Figure 6: Schematic representation of the protocol for generation of materials using the decoder part of the trained CVAE model.

Training

Keras[61] deep learning framework with Tensorflow[62] back-end was used for all the training and validation purposes. Mean absolute error was used as the objective function to train the the predictor networks. For training the generative model, categorical cross entropy was used as the objective function, and Kullback-Leibler (KL) divergence loss was used to ensure that encoded samples in the latent space followed a normal distribution. ADAM optimizer[63] with the parameters as suggested by the authors was used for both the networks. The learning rate[64] was reduced on plateau with a patience of 5 epochs from 0.1 to 10^{-5} during the training process of both the networks.

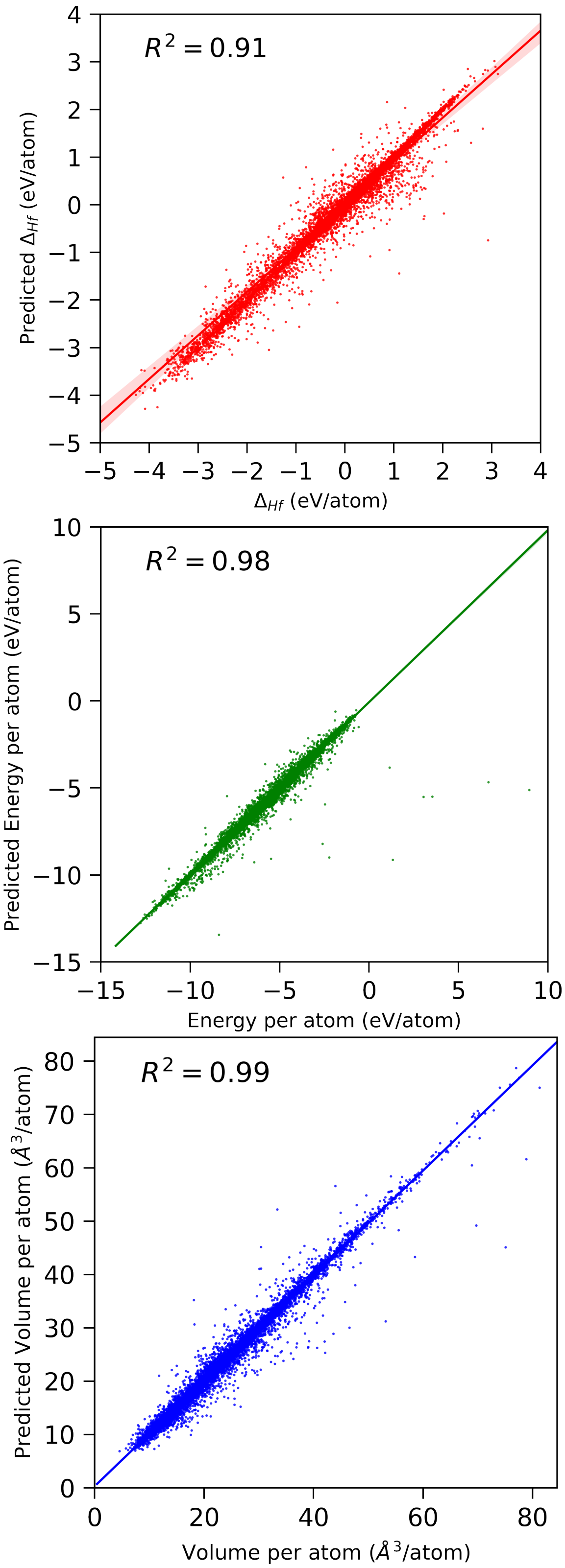


Figure 7: Plots of the correlation between the predicted and ground truth values of the three properties calculated for the materials in the test set. Linear fit and the coefficient of determination are also depicted.

Results and Discussion

The framework proposed in this study consists of two parts: one is the generator network and second is the predictor networks. The generator network, which is based on CVAE,[52] is used to propose material candidates with desired target properties. The predictor networks are used to assess the proposed candidates and filter/validate the generated candidates. For this method to be accurate, it is ideal that the generator network proposes more candidates in the region of desired target, and the predictor networks are able to accurately assess the property values of candidates in this region. In this section, the accuracy of the predictor models followed by the ability of the DING model to generate materials with desired properties are presented.

Three independent neural networks each for the predictions of formation enthalpy, volume per atom and energy per atom that were trained on the OQMD dataset[12] with a similar architecture comprise the predictor module (Figure 4 and Table 1). Figure 7 depicts the correlation between the three predicted properties considered in this study and the DFT calculated values corresponding to the test set of the dataset. The mean absolute errors for the prediction of formation enthalpy, volume per atom and energy per atom are 0.051 eV/atom, 0.398 Å^3/atom and 0.072 eV/atom respectively. The coefficient of determination corresponding to these plots are also given in the plots. These two measures indicate that the predictor module is able to predict all the three properties considered here reasonably accurately.

Though the predictor module is very accurate in predicting the three material properties, it is clear from Figure 7 that predictions in certain regions are less accurate. Hence the predictor module is not applicable in those ranges. For example, when the DFT calculated energy per atom value is more than –0.5 eV, the performance of the predictor model is not adequate. This is possibly because of the availability of less samples for satisfactory training. This problem is less pronounced in the other two predictors which model the formation enthalpy and the volume per atom. A qualitative analysis of the errors in the predictions

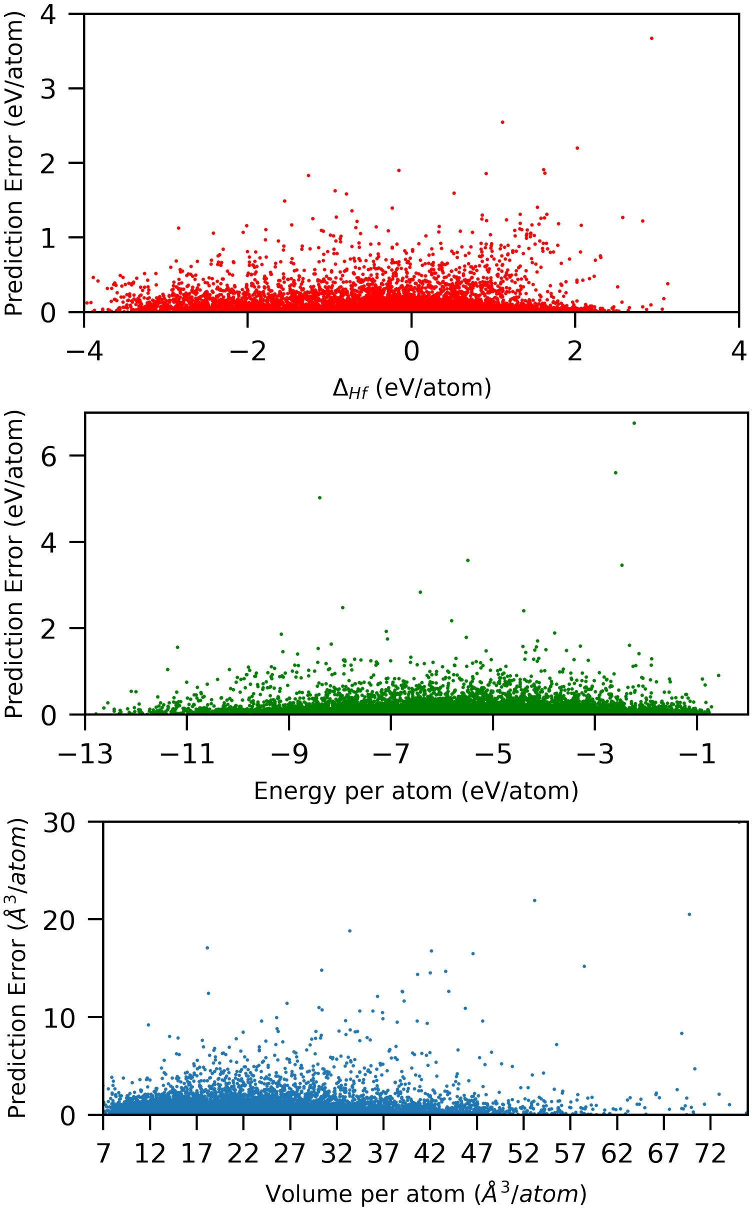


Figure 8: The absolute errors of the predictor models in the range of material property values where the applicability of the model is adequate.

was done and the range of values that are reliably predicted by the model was identified (see Table 3). The absolute error of the values as a function of the ground truth values of the three properties are given in Figure 8. The mean absolute errors corresponding to these ranges are also given in Table 3. Notably, these MAEs are close to those given above, which emphasizes that this exercise is done to identify the range of target values where the model is applicable.

Table 3: The ranges of the property values within which the predictor module is applicable and accurate. The MAE in the given range is also given.

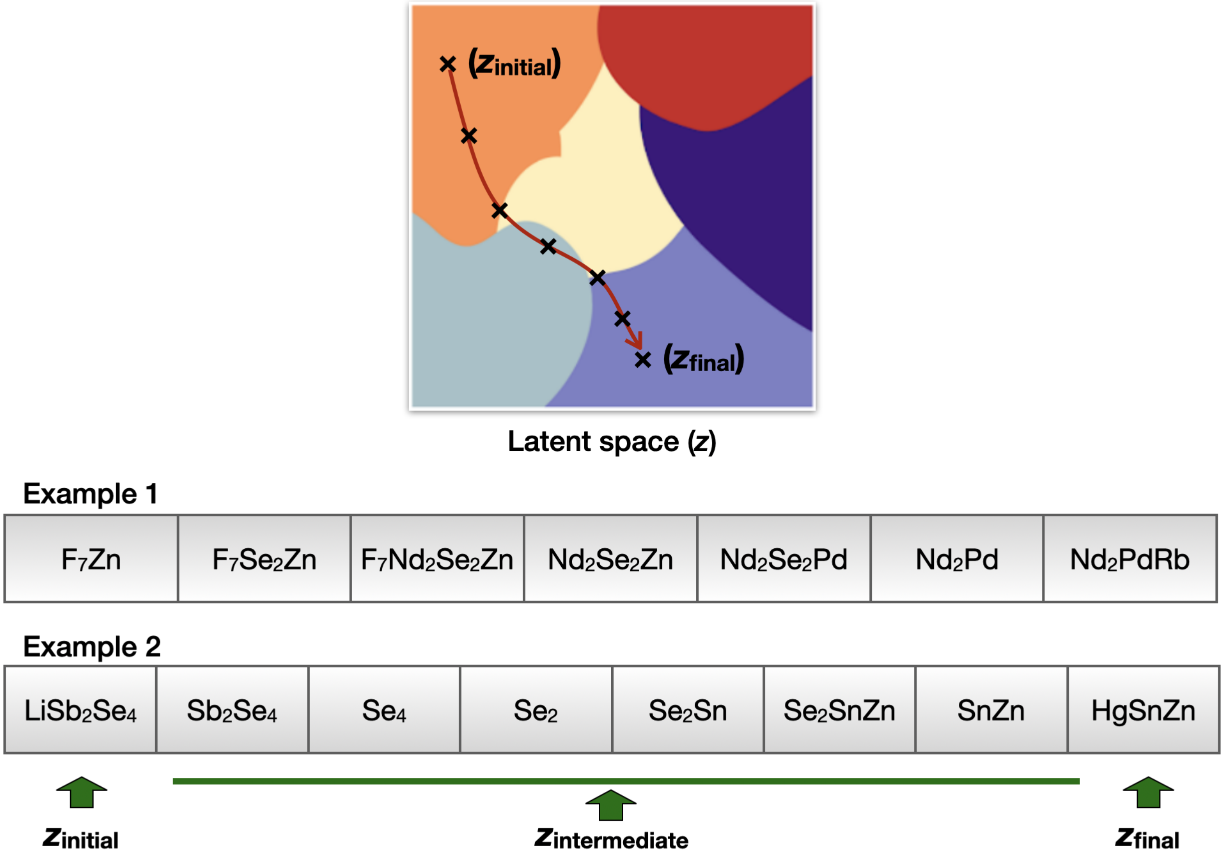
Property	Minimum	Maximum	Mean Absolute Error
Formation Enthalpy	–3.90 eV/atom	3.06 eV/atom	0.048 eV/atom
Energy per atom	–12.78 eV/atom	–0.77 eV/atom	0.069 eV/atom
Volume per atom	7.06 Å^3/atom	75.83 Å^3/atom	0.397 Å^3/atom

The generator module was built using CVAE, which is responsible for the generation of candidate material compositions with target property values. One of the primary aspects of the CVAE model is its ability to reconstruct the input features with high accuracy.[52] After training the network using categorical cross entropy along with KL divergence as the objective function, the model was able to reconstruct $94.30 \pm 0.36\%$ materials from the test set averaged across 5 independent runs. In other words, after training, the input and the output feature vectors were identical for $94.30 \pm 0.36\%$ of the materials in the test set. This indicates that the 32-dimensional latent space (bottleneck) is able to capture most of the material features originally represented in 979-dimensions.

The second important aspect of the CVAE model relevant to material generation is the continuity of the latent space. Such a continuous latent space is differentiable making gradient based optimization for searching the material space possible.[65,66] It is also possible to interpolate between two points on the continuous representations/materials[48] and also allows for decoding random vectors in the latent space to material compositions. In the trained DING model, a walk between latent vectors of two generated materials with the same properties was undertaken to examine the continuous nature of the latent space (Figure 9). This is done by comparing the material compositions of the initial and final with that of the intermediate points along the walk. Once the initial and final points corresponding to the two generated materials on the latent space are chosen, the following equation is used to generate the intermediate latent vectors.

$$z_{intermediate} = z_{initial} + (z_{final} - z_{initial}) * \epsilon \tag{3}$$

Here $z_{initial}$ and z_{final} corresponds to the latent vector of the two generated materials and $z_{intermediate}$ corresponds to the latent vector of the intermediate materials while taking the walk. The initial value of ϵ is 0 and was incremented by 0.01 at each step until the material corresponding to z_{final} is generated.



Example 1

F_7Zn	F_7Se_2Zn	$F_7Nd_2Se_2Zn$	Nd_2Se_2Zn	Nd_2Se_2Pd	Nd_2Pd	Nd_2PdRb

Example 2

$LiSb_2Se_4$	Sb_2Se_4	Se_4	Se_2	Se_2Sn	Se_2SnZn	SnZn	HgSnZn

Figure 9: Schematic representation of the initial and final points on the latent space and the path taken from $z_{initial}$ and z_{final} (top). Two examples of the initial, final and intermediate points corresponding to generated materials are also given (bottom).

Figure 9 depicts the schematic representation of the initial and final points on the latent space corresponding to two materials ($z_{initial}$ and z_{final}). The path connecting the two points via intermediate material compositions is given by a red arrow. If the latent space obtained in the DING model is truly continuous, it should in someway reflect in the material compositions. Two examples were chosen to demonstrate this phenomenon: (i) F_7Zn and Nd_2PdRb as initial and final material compositions, and (ii) $LiSb_2Se4$ and HgSnZn as the initial and final material compositions. The intermediates going from these two initial states to their corresponding final states exhibit a smooth transition. Any two adjacent materials in the path involves only a single element being added or removed when going from one to the other in either direction substantiating the continuous nature of the latent space.

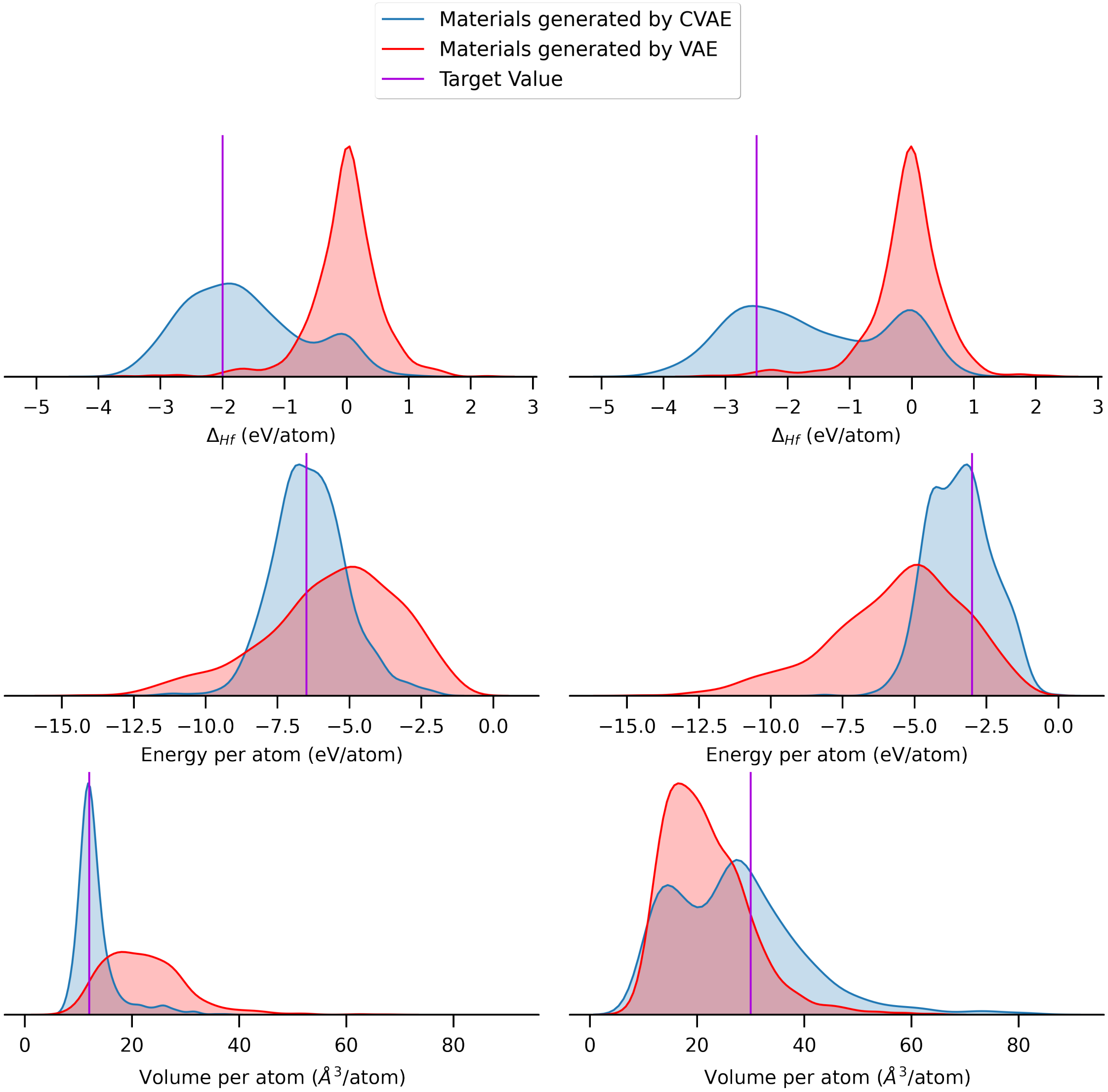


Figure 10: Probability distributions of the properties of the materials generated with two sets of target values using the trained CVAE (DING) and VAE models. The target values of each properties are given in purple spikes along with the corresponding distributions.

The above sections demonstrated that the CVAE based DING model is able to accurately reconstruct the input vectors and the significantly lower dimensional latent space is continuous. The third aspect of the DING model that needs examination is its ability to generate material compositions that exhibit certain target values of material properties. The decoder network of the trained CVAE model was used for the generation of new material candidates.

A random z sampled from $N(0, I)$ along with the desired target property is used as the input for the decoder network. The output of the decoder network is the feature vector corresponding to the generated new material candidate, which can further be examined using the predictor networks discussed above. As a baseline, a variational autoencoder (VAE)[48,66] model was trained on the same training set. The reconstruction accuracy of this trained VAE model is $93.72 \pm 0.60\%$ for the test set averaged across 5 independent runs. The target property is selected from the acceptable range defined earlier for the predictor models (Figure 8 and Table 3). Comparison of the properties of material candidates generated by the the trained CVAE model with this VAE model is expected to reveal the capability of DING framework to generate material compositions with desired properties.

The DING decoder and the trained VAE model were used to generated 1000 unique material compositions each based on different sets of target property values. Two select set of values of the three properties (–2 ev/atom, –2.5 ev/atom & –6.5 ev/atom, –3 ev/atom & 12 Å^3/atom, 30 Å^3/atom for formation enthalpy, energy per atom and volume per atom respectively) are chosen here to demonstrate the generation capability of the DING method. The predictor module was used to calculate the three properties for all the 4000 material compositions whose probability distributions are depicted in Figure 10. As expected, the probability distributions of the properties of the materials generated using the VAE model roughly resembles the distribution of the dataset chosen for the training. However, the distributions of the material properties obtained based on the DING model shift towards the target values (indicated by a purple colored spike) in every instance. This clearly demonstrates that it is possible to manipulate the continuous latent space obtained in the DING model to generate materials with desired properties. In other words, DING model allows for focused searching of materials in an otherwise large chemical space.

Conclusion

In this study, a deep learning model, namely DING, developed using CVAE capable of generating candidate materials with desired properties is reported. Such a method is significantly more efficient than the conventional combinatorial search approach both in terms of time and in terms of comprehensive searching of the chemical space. A novel feature vector, which effectively encodes the composition of the materials is used here based on which both the predictor and generator modules were trained. The predictor models part of the DING framework, which are deep neural networks, exhibit high accuracy demonstrating the robustness of the trained network and the adequacy of the feature vector. The generator module consists of a latent space that is crucial for the generation of new materials. Examination of the continuous nature of the latent space indicates that the model has learnt the underlying chemistry in terms of the material composition. The predictor module was used to examine the materials generated with desired set of properties, and it clearly indicates that the generator is able to positively bias the material search in the chemical space. This method is successful in accurately predicting material properties and is able to generate new material candidates; however, it is desirable that model is able to learn and generate three-dimensional structures of the materials than just the material compositions. Future work in this direction is in progress.

Acknowledgement

We thank DST-SERB (grant no. EMR/2016/007697) for financial assistance.

References

(1) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.

(2) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine learning for molecular and materials science. *Nature* **2018**, *559*, 547.

(3) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep learning for computational chemistry. *J. Comput. Chem.* **2017**, *38*, 1291–1307.

(4) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59*, 2545–2559.

(5) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360–365.

(6) Webb, S. Deep learning for biology. *Nature* **2018**, *554*.

(7) Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W.; Bridgland, A., et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, 1–5.

(8) Popova, M.; Isayev, O.; Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **2018**, *4*, eaap7885.

(9) Segler, M. H.; Preuss, M.; Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604.

(10) Y. Pathak, S. M., S. Laghuvarapu; Priyakumar, U. D. Chemically Interpretable Graph Interaction Network for Prediction of Pharmacokinetic Properties of Drug-like Molecules. *ChemRxiv Preprint* **2019**, https://doi.org/10.26434/chemrxiv.10282346.v1.

(11) Laghuvarapu, S.; Pathak, Y.; Priyakumar, U. D. Band nn: A deep learning framework for energy prediction and geometry optimization of organic small molecules. *J. Comput. Chem.* **2020**, *41*, 790–799.

(12) Kirklin, S.; Saal, J. E.; Meredig, B.; Thompson, A.; Doak, J. W.; Aykol, M.; Rühl, S.; Wolverton, C. The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies. *npj Comput. Mater.* **2015**, *1*, 15010.

(13) Saal, J. E.; Kirklin, S.; Aykol, M.; Meredig, B.; Wolverton, C. Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD). *JOM* **2013**, *65*, 1501–1509.

(14) Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G., et al. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater.* **2013**, *1*, 011002.

(15) Curtarolo, S.; Setyawan, W.; Wang, S.; Xue, J.; Yang, K.; Taylor, R. H.; Nelson, L. J.; Hart, G. L.; Sanvito, S.; Buongiorno-Nardelli, M., et al. AFLOWLIB. ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.* **2012**, *58*, 227–235.

(16) Irwin, J. J.; Shoichet, B. K. ZINC- a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **2005**, *45*, 177–182.

(17) Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864–2875.

(18) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E., et al. The ChEMBL database in 2017. *Nucleic Acids Res.* **2017**, *45*, D945–D954.

(19) Srinivasan, S.; Rajan, K. “Property phase diagrams” for compound semiconductors through data mining. *Materials* **2013**, *6*, 279–290.

(20) Jha, D.; Ward, L.; Paul, A.; Liao, W.-k.; Choudhary, A.; Wolverton, C.; Agrawal, A. ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition. *Sci. Rep* **2018**, *8*, 17593.

(21) Ghiringhelli, L. M.; Vybiral, J.; Levchenko, S. V.; Draxl, C.; Scheffler, M. Big data of materials science: critical role of the descriptor. *Phys. Rev. Lett.* **2015**, *114*, 105503.

(22) Meredig, B.; Agrawal, A.; Kirklin, S.; Saal, J. E.; Doak, J. W.; Thompson, A.; Zhang, K.; Choudhary, A.; Wolverton, C. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Phys. Rev. B* **2014**, *89*, 094104.

(23) Kong, C. S.; Luo, W.; Arapan, S.; Villars, P.; Iwata, S.; Ahuja, R.; Rajan, K. Information-theoretic approach for the discovery of design rules for crystal chemistry. *J. Chem. Inf. Model.* **2012**, *52*, 1812–1820.

(24) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Crystal structure representations for machine learning models of formation energies. *Int. J. Quantum Chem.* **2015**, *115*, 1094–1101.

(25) Schütt, K.; Glawe, H.; Brockherde, F.; Sanna, A.; Müller, K.; Gross, E. How to represent crystal structures for machine learning: Towards fast prediction of electronic properties. *Phys. Rev. B* **2014**, *89*, 205118.

(26) Pilania, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. Accelerating materials property predictions using machine learning. *Sci. Rep* **2013**, *3*, 2810.

(27) Ward, L.; Agrawal, A.; Choudhary, A.; Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Comput. Mater.* **2016**, *2*, 16028.

(28) Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.; Kim, C. Machine

learning in materials informatics: recent applications and prospects. *npj Comput. Mater.* **2017**, *3*, 54.

(29) Fernandez, M.; Boyd, P. G.; Daff, T. D.; Aghaji, M. Z.; Woo, T. K. Rapid and accurate machine learning recognition of high performing metal organic frameworks for CO2 capture. *J. Phys. Chem. Lett.* **2014**, *5*, 3056–3060.

(30) Oliynyk, A. O.; Antono, E.; Sparks, T. D.; Ghadbeigi, L.; Gaultois, M. W.; Meredig, B.; Mar, A. High-throughput machine-learning-driven synthesis of full-Heusler compounds. *Chem. Mater.* **2016**, *28*, 7324–7331.

(31) Isayev, O.; Oses, C.; Toher, C.; Gossett, E.; Curtarolo, S.; Tropsha, A. Universal fragment descriptors for predicting properties of inorganic crystals. *Nat. Commun.* **2017**, *8*, 15679.

(32) Raccuglia, P.; Elbert, K. C.; Adler, P. D.; Falk, C.; Wenny, M. B.; Mollo, A.; Zeller, M.; Friedler, S. A.; Schrier, J.; Norquist, A. J. Machine-learning-assisted materials discovery using failed experiments. *Nature* **2016**, *533*, 73.

(33) Stanev, V.; Oses, C.; Kusne, A. G.; Rodriguez, E.; Paglione, J.; Curtarolo, S.; Takeuchi, I. Machine learning modeling of superconducting critical temperature. *npj Comput. Mater.* **2018**, *4*, 29.

(34) Sun, X.; Hou, Z.; Sumita, M.; Ishihara, S.; Tamura, R.; Tsuda, K. Data Integration for Accelerated Materials Design via Preference Learning. *New J. Phys.* **2020**, *22*, 055001.

(35) Saito, Y.; Shin, K.; Terayama, K.; Desai, S.; Onga, M.; Nakagawa, Y.; Itahashi, Y. M.; Iwasa, Y.; Yamada, M.; Tsuda, K. Deep-learning-based quality filtering of mechanically exfoliated 2D crystals. *npj Comput. Mater.* **2019**, *5*, 1–6.

(36) Ward, L.; Wolverton, C. Atomistic calculations and materials informatics: A review. *Curr Opin Solid State Mater Sci.* **2017**, *21*, 167–176.

(37) Kohn, W. Nobel Lecture: Electronic structure of matter—wave functions and density functionals. *Rev. Mod. Phys.* **1999**, *71*, 1253.

(38) Hafner, J.; Wolverton, C.; Ceder, G. Toward computational materials design: the impact of density functional theory on materials research. *MRS bulletin* **2006**, *31*, 659–668.

(39) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276.

(40) Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; Aspuru-Guzik, A. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *ChemRxiv Preprint* **2017**, https://doi.org/10.26434/chemrxiv.5309668.v3.

(41) De Cao, N.; Kipf, T. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973* **2018**,

(42) You, J.; Liu, B.; Ying, Z.; Pande, V.; Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. Advances in Neural Information Processing Systems. 2018; pp 6410–6421.

(43) Kingma, D. P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114* **2013**,

(44) Noh, J.; Kim, J.; Stein, H. S.; Sanchez-Lengeling, B.; Gregoire, J. M.; Aspuru-Guzik, A.; Jung, Y. Inverse Design of Solid-State Materials via a Continuous Representation. *Matter* **2019**, *1*, 1370–1384.

(45) Hoffmann, J.; Maestrati, L.; Sawada, Y.; Tang, J.; Sellier, J. M.; Bengio, Y. Data-

Driven Approach to Encoding and Decoding 3-D Crystal Structures. *arXiv preprint arXiv:1909.00949* **2019**,

(46) Nouira, A.; Sokolovska, N.; Crivello, J.-C. CrystalGAN: Learning to Discover Crystallographic Structures with Generative Adversarial Networks. *arXiv preprint arXiv:1810.11203* **2018**,

(47) Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv preprint arXiv:1406.2661* **2014**,

(48) Kingma, D. P.; Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**,

(49) Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602* **2013**,

(50) Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.* **2019**, *10*, 3567–3572.

(51) Brown, N.; McKay, B.; Gilardoni, F.; Gasteiger, J. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1079–1087.

(52) Sohn, K.; Lee, H.; Yan, X. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc., 2015; pp 3483–3491.

(53) Kingma, D. P.; Rezende, D. J.; Mohamed, S.; Welling, M. Semi-Supervised Learning with Deep Generative Models. *arXiv preprint arXiv:1406.5298* **2014**,

(54) Mishra, A.; Krishna Reddy, S.; Mittal, A.; Murthy, H. A. A generative model for zero shot learning using conditional variational autoencoders. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018; pp 2188–2196.

(55) Walker, J.; Doersch, C.; Gupta, A.; Hebert, M. An uncertain future: Forecasting from static images using variational autoencoders. European Conference on Computer Vision. 2016; pp 835–851.

(56) Zhao, T.; Zhao, R.; Eskenazi, M. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960* **2017**,

(57) Shen, X.; Su, H.; Li, Y.; Li, W.; Niu, S.; Zhao, Y.; Aizawa, A.; Long, G. A conditional variational framework for dialog generation. *arXiv preprint arXiv:1705.00316* **2017**,

(58) Lim, J.; Ryu, S.; Kim, J. W.; Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J. Chem. Inf.* **2018**, *10*, 31.

(59) Bergerhoff, G.; Hundt, R.; Sievers, R.; Brown, I. D. The inorganic crystal structure data base. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 66–69.

(60) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res* **2014**, *15*, 1929–1958.

(61) Chollet, F., et al. Keras. `https://keras.io`, 2015.

(62) Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; `http://tensorflow.org/`, Software available from tensorflow.org.

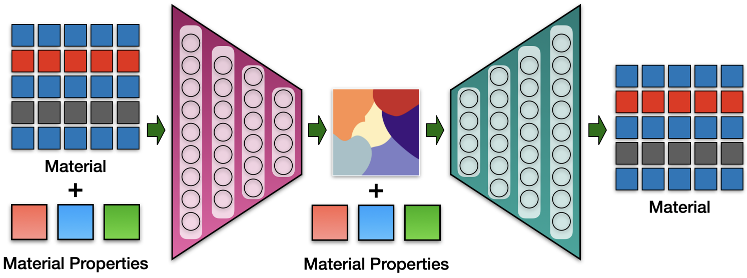
(63) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(64) Jacobs, R. A. Increased rates of convergence through learning rate adaptation. *Neural networks* **1988**, *1*, 295–307.

(65) Doersch, C. Tutorial on Variational Autoencoders. *arXiv preprint arXiv:1606.05908* **2016**,

(66) Kingma, D. P.; Welling, M. An Introduction to Variational Autoencoders. *Found. Trends Mach. Learn.* **2019**, *12*, 307–392.

Graphical TOC Entry



DING: **D**eep **IN**organic material **G**enerator