

## ***De Novo* Design of Molecules Towards Biased Properties via a Deep Generative Framework and Iterative Transfer Learning**

Kianoosh Sattari<sup>1</sup>, Dawei Li<sup>1</sup>, Yunchao Xie<sup>1</sup>, Olexandr Isayev<sup>4</sup>, and Jian Lin<sup>1, 2, 3\*</sup>

<sup>1</sup>Department of Mechanical and Aerospace Engineering,

<sup>2</sup>Department of Electrical Engineering and Computer Science,

<sup>3</sup>Department of Physics and Astronomy,

University of Missouri, Columbia, Missouri 65211, United States

<sup>4</sup>Department of Chemistry, Carnegie Mellon University, Pittsburgh, PA 15213, United States

\*E-mail: LinJian@missouri.edu (J. L.)

### **Abstract**

*De Novo* design of molecules with targeted properties represents a new frontier in molecule development. Despite enormous progress, two main challenges remain, i.e., (i) generation of novel molecules with targeted and quantifiable properties; (ii) generated molecules having property values beyond the range in the training dataset. To tackle these challenges, we propose a novel reinforced regressional and conditional generative adversarial network (RRCGAN) to generate chemically valid, drug-like molecules with targeted heat capacity ( $C_v$ ) values as a proof-of-concept study. As validated by DFT, ~80% of the generated samples have a relative error (RE) of < 20% of the targeted  $C_v$  values. To bias the generation of molecules with the  $C_v$  values beyond the range of the original training molecules, transfer learning was applied to iteratively retrain the RRCGAN model. After only two iterations of transfer learning, the mean  $C_v$  of the generated molecules increases to 44.0 cal/(mol·K) from the mean value of 31.6 cal/(mol·K) shown in the initial training dataset. This demonstrated computation methodology paves a new avenue to discovering drug-like molecules with biased properties, which can be straightforwardly repurposed for optimizing individual or multi-objective properties of various matters.

**Keywords:** inverse design, molecules, reinforced regressional and conditional generative adversarial network, targeted properties, transfer learning

## 1. Introduction

Discovery of novel drug-like molecules with superior properties advances human health. Success has been demonstrated by developing vaccines and drugs for the treatment of global COVID-19 pandemic. To develop new molecules, a stepwise procedure of molecule design, property prediction, chemical synthesis, and experimental evaluation is usually repeated until satisfactory performance is achieved. Despite much progress in the past decades, this task remains challenging due to two main reasons. First, the massive, discrete, and unsaturated design space ( $\sim 10^{60}$ ) makes the traditional experimental and computational approaches impractical to fully explore the entire space.<sup>1,2</sup> Second, a slight change in a molecule structure can radically change its properties, making the on-demand molecule design even more difficult.<sup>3</sup> Hence, a new research paradigm is quite desired to expedite the process. High-throughput virtual and experimental screening (HTVS and HTES) methods have emerged to accelerate molecule discovery in the past three decades.<sup>4</sup> They iteratively generate, synthesize, and evaluate the molecules from an enormous library of molecular fragments by combinatorial enumeration.<sup>4-8</sup> Although they accelerate examination of the design space by 3-5 orders of magnitude, their coverage and success rate are still far from the need of discovering a sufficient number of novel molecules.<sup>4</sup> In addition to HTVS and HTES, global optimization (GO) strategies such as genetic algorithms have made much progress in identifying the top-ranked molecules,<sup>9,10</sup> since they can efficiently screen the molecules with high-ranking scores from a fraction of possible candidates. However, the GO strategies require prior rules on how to transform the molecular fragments, thus greatly restricting the number of molecules to be explored. Moreover, the accuracy dramatically decreases as the system complexity increases.<sup>11</sup> Finally, many evolution steps are required to obtain the optimal

candidates, thus making them not suitable for the on-demand generation of novel molecules with targeted properties.

Recently, machine learning (ML) algorithms, particularly deep learning (DL), have been applied to discover novel molecules since they can learn hidden knowledge from a large scale of data.<sup>12</sup> They are used for predicting properties of the materials. For instance, they have been widely implemented to assist or even substitute theoretical simulations in HTVS of molecules in photovoltaics,<sup>13</sup> photocatalysis,<sup>14</sup> and antimicrobial applications.<sup>15</sup> They are also applied as generative models (GMs) for inverse molecule design. A GM-based inverse design process starts with mapping the high-dimensional representations of the molecules to a reduced latent space, which is then used to search for or optimize new molecules. They can identify hidden patterns from the highly complex, nonlinear data in an automatic and on-demand fashion without much prior knowledge for creating non-intuitive, even counterintuitive molecules that outperform the empirically designed ones. Thus, they are well suited for exploratory optimization problems in the unrestricted design space. For instance, variational Autoencoders (VAEs),<sup>16</sup> generative adversarial networks (GANs),<sup>17</sup> and reinforcement learning (RL),<sup>18,19</sup> and recurrent neural networks (RNNs),<sup>20,21</sup> or integration of these networks into a new architecture, have made the inverse molecule design become more and more feasible.<sup>22,23</sup> Aspuru-Guzik et al. employed a VAE model to map discrete representations of molecules to continuous ones for automatically generating novel molecules, which makes the gradient-based search of a chemical space possible.<sup>3</sup> However, the model is not suitable for *de novo* generation of molecules with targeted, quantifiable properties. Zhavoronkov and coworkers trained a generative adversarial autoencoder (AAE) using molecular fingerprints and drug concentrations as the input features to generate molecules,<sup>24</sup> while it is costly and requires hand-crafted rules to avoid impractical molecules. An RNN model was proposed to

generate molecular libraries with targeted bioactivities but resulted in inaccurate control over the properties of the generated molecules compared to the targeted ones.<sup>20</sup> Popova et al. proposed an RNN-based generative model within an RL framework to generate compounds with targeted melting temperatures.<sup>21</sup> It generates the compounds with properties following those of the training samples, but it is still not on-demand generation upon the targeted property values. Such on-demand generation algorithms are necessary for moving toward autonomous laboratories.<sup>25</sup>

Among the various DL models proposed recently for molecule design, GANs bring in breakthroughs. Two proof-of-concept GANs, namely ORGAN<sup>26</sup> and ORGANIC,<sup>27</sup> were introduced to generate novel molecules, while the generation is not conditioned on the physicochemical or biological properties with quantitative and continuous labels. Our group recently proposed a regressional and conditional GAN (RCGAN) for the inverse design of two-dimensional structures.<sup>11</sup> RCGAN can meet two criteria for inversely designing new structures: 1) generating distinguished structures from the training samples; 2) generating structures conditioned on the input quantitative, continuous labels. However, all these past GAN-based models generate the structures with the targeted properties in the range of the training samples, or so-called interpolation. To the best of our knowledge, a GM model that can perform an extrapolation task for the generation of molecules with targeted properties beyond the limit value of the training dataset has been rarely reported, if not any.

To tackle this challenge, we propose a new computational framework that integrates a novel reinforced RCGAN (RRCGAN) architecture. RRCGAN consists of three networks with a transfer learning algorithm to iteratively update RRCGAN for the targeted generation of molecules with quantitative, continuous property values beyond the initial training dataset. RRCGAN includes an autoencoder (AE), an RCGAN network, and a reinforcement center. AE encodes discrete

representations of the molecules to continuous latent features, which are then fed as the input to RCGAN. RCGAN includes regressor, generator, and discriminator networks. The reinforcement center biases RCGAN towards generating valid and accurate molecules. Thus, the architecture is named RRCGAN. Transfer learning was implemented to iteratively retrain RRCGAN for generating new molecules showing increased property values compared to those of initial training data. As a proof of concept, we employed RRCGAN to generate realistic molecules with biased isochoric heat capacity ( $C_v$ ), because  $C_v$  is a fundamental thermodynamical property.<sup>28</sup> In this work, we started by training the RRCGAN model with ~63 thousand molecules whose  $C_v$  are normally distributed from 20.9 to 42.3 cal/(mol K) after outliers were removed from the original QM9 library. Then, it was transferred to a new model that was finetuned on ~2000 molecules with  $C_v$  of > 42.3 cal/(mol K) from the original QM9 library as well as outliers of the generated molecules by the original model. This first iterated transfer learning model generated new molecules with increased  $C_v$ , which were validated by DFT and then used to retrain the second iterated transfer learning model to generate new molecules with even higher  $C_v$ . The novelty of this iterative searching algorithm can be summarized as follows. First, the generated molecules meet the novelty and diversity in the molecular structures and have high chemical validity and accuracy. Second, the generated molecules have the targeted properties in continuous, quantitative labels. Third, the generation is purely data-driven and can be extrapolated beyond the range of the initial training dataset by transfer learning. Last but not least, the model is robust and can be updated for generating molecules that meet multiple targeted properties or other matter with biased properties.

**Commented [SK(S1)]:** I recently revisited the model, I used ~2000 molecules for each iteration.

**Commented [lj2R1]:** Did you update the results and figures? Did you update them the Methods and SI?

## 2. Results and Discussion

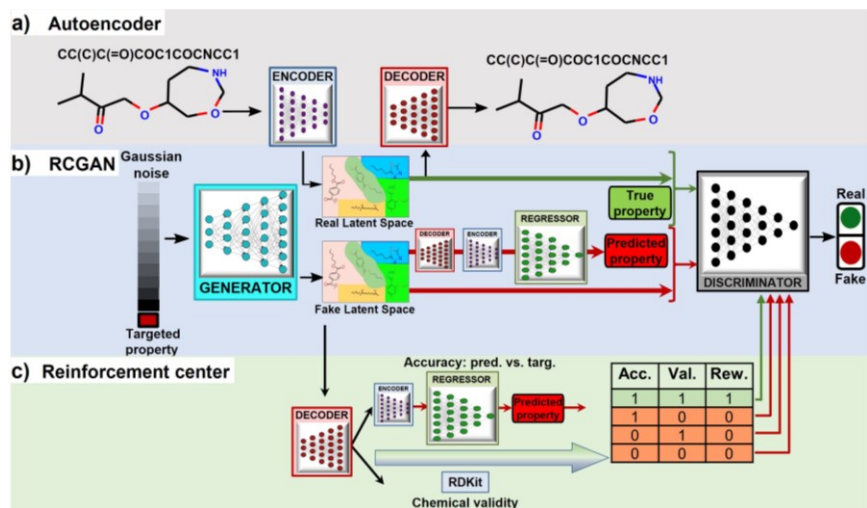
## 2.1 Development of RRCGAN

**Architecture of RRCGAN.** Fig. 1 represents the schematic of the RRCGAN architecture, which includes AE, RCGAN, and a reinforcement center. All the initial training molecules are collected from the QM9 library<sup>29</sup> and represented by the simplified molecular-input line-entry system (SMILES) strings (Supplementary Note 1).<sup>30</sup> Atom and bond information of the molecules is one-hot encoded in these SMILES strings (Fig. S1). Details about the SMILES representations can be found in Supplementary Note 1. AE consists of an encoder and a decoder (Fig. 1a). The encoder maps the discrete molecular representations to continuous latent vectors, while the decoder converts the continuous vectors back to the discrete representations.<sup>30</sup> It is trained to minimize the error in reproducing the original SMILES strings. The encoder is a convolutional neural network (CNN) (Fig. S2). It outputs fixed-dimensional latent vectors ( $6 \times 6 \times 2$ ) that have the most statistically important information from the input SMILES strings. The architecture of the decoder was modified from Google Inception V2 (Fig. S3).<sup>31</sup> The decoder converts the latent vectors back to the original SMILES strings (input to the encoder).

In this work, RCGAN has a generator, a discriminator, and a regressor network. RCGAN learns the hidden relationship between the latent vectors and properties of the training molecules for generating new latent vectors conditioned on targeted  $C_v$  (continuous, quantified labels), which are then converted to SMILES using the decoder (Fig. 1b). The regressor was modified from Google Inception V2<sup>31</sup> (Fig. S4). It was built as a quantitative structure-activity relationship (QSPR) model for predicting  $C_v$ . To generate a latent vector conditioned on a  $C_v$  value, the generator receives a concatenated vector consisting of a targeted  $C_v$  and a randomly generated noise vector  $z$  with a dimension of  $128 \times 1$  (Fig. S5). The generated latent vector has a dimension of  $6 \times 6 \times 2$  and is expected to contain chemical information hidden in the high-dimensional training

data. To train the discriminator, a concatenated vector that consists of the latent vector synthesized by the generator and a  $C_v$  value predicted by the regressor is fed into the discriminator (Fig. S6). The trained encoder is used to convert the synthesized latent vectors to SMILES that are then fed into the trained encoder to generate the latent vectors, which serve as the input to the regressor. The trained regressor then predicts the property that corresponds to the generated latent vectors. If the regressor is directly fed with the generated latent vectors, the prediction is not accurate. The reason is that the latent vectors should be mapped to the same space as the real latent vectors (Fig. 1b). The discriminator performs two functions. First, it determines whether the concatenated vector is from a real molecule or a fake one by comparing the statistical distribution of the two. Second, it tells whether a generated molecule corresponds to the targeted  $C_v$  value.

Finally, a reinforcement center is included in RRCGAN to ensure that the generated molecules are chemically valid and accurate in comparison of the validated  $C_v$  with the targeted  $C_v$  (Fig. 1c). First, the latent vectors generated by the generator are converted to the SMILES by the decoder and then fed into RDKit<sup>32</sup> to ensure that the SMILES are chemically valid. If a SMILES is valid, then “1” is assigned to the string; otherwise, “0” is assigned. Subsequently, a relative error (RE) of a targeted  $C_v$  compared with the predicted value from the regressor is evaluated. If RE is less than 20%, then “1” is assigned. Only a SMILES with assigned numbers of both “1” is labeled as a real sample. Otherwise, it is labeled as a fake one. These two constraints reinforce the discriminator to consider the molecules with both high chemical validity and high accuracy as real and others as fake. In the training process, before the reinforcement center is activated, the generator and the discriminator are trained with an initial number of epochs. Details about the architectures of these networks and their training processes are described in Supplementary Note 2. Their loss functions and training processes are described as follows.



**Figure 1.** The schematic of the proposed RRCGAN for inverse design of molecules with **targeted  $C_v$** . (a) Architecture of the autoencoder for converting discrete representations of molecules to and from a continuous latent space. (b) Architecture of the RCGAN framework. The generator takes targeted property and Gaussian noise as inputs to generate latent vectors. The discriminator distinguishes fake samples from the real ones based on their latent vectors and their assigned properties. The regressor predicts the property values from the generated latent vectors. (c) Scheme of the reinforcement center that biases the discriminator towards generation of the valid and accurate generated samples.

**Loss functions of encoder, decoder, regressor, generator, and discriminator.** The loss function ( $L_{AE}$ ) of AE is the sum of the cross entropy ( $L_{AE1}$  for discrete one-hot encoded SMILES strings) and the mean square error (MSE) ( $L_{AE2}$  for continuous property of interest), as shown in the following equations.



$$L_{AE} = L_{AE1} + L_{AE2} \quad (1)$$

$$L_{AE1} = \sum_i^N -t_i \log(\hat{t}_i) - (1-t_i) \log(1-\hat{t}_i) \quad (2)$$

$$L_{AE2}[Y, Y] = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3)$$

In Eq. 2,  $t$  is the true value (either 0 or 1) showing binary categories in the one-hot encoding vectors used for each SMILES. The predicted  $\hat{t}$  can be any value between zero and one, while they must sum to 1 in the last SoftMax layer of the decoder. In Eq. 3,  $\hat{y}$  is the predicted  $C_v$ ,  $y$  is the true  $C_v$ , and  $N$  is the number of samples.

The loss function of the regressor is defined as the  $L_2$  in Eq. 4. It measures the difference between the predicted and targeted  $C_v$ .

$$Loss_R = L_2[Y, R(Z)] \quad (4)$$

where  $Z$  is a latent vector output from the encoder and  $Y$  is the targeted  $C_v$ .  $L_2$  is defined in Eq. 5.

$$L_2[Y, Y] = \frac{1}{N} \sum_i (y_i - \hat{y}_i)^2 \quad (5)$$

where  $y$  is the targeted  $C_v$  value,  $\hat{y}$  is the predicted  $C_v$  value from the regressor or  $R(Z)$ , and  $N$  is the number of samples.

The loss function of the generator as shown in Eq. 6. It includes two terms. The first one is the same as the loss function for the least square GAN (LSGAN),<sup>33</sup> while the second one is the regularized loss for the regressor.

$$Loss_G = \frac{1}{2} \mathbb{E}_{z \sim P_z(z)} \left[ D \left( (G(z, Y)), Y \right) - 1 \right]^2 + w L_2 \left( Y, R(E(D_2(G(z, Y)))) \right) \quad (6)$$

where  $\mathbb{E}$  is the expectation function, the subscript  $(z \sim P_z(z))$  shows the synthesized molecules from the generator, and  $z$  is a random noise and the input of the generator.  $D_2$  and  $E$  are the decoder and encoder, respectively.  $D$  is the discriminator that uses the latent vectors generated from the

generator and the predicted  $C_v$  from the regressor to classify them into two groups of fake [0] or real [1] samples. When feeding the regressor with the generated molecules, the  $L_2$  loss is calculated and then used as the regularization term in the loss function of the generator.  $w$  is the weighting parameter for the regularization term. The combined loss function ensures that the generator and discriminator are simultaneously trained to avoid mode collapse.

The loss function of the discriminator is the same as the one used for LSGAN (Eq. 7).<sup>33</sup>

$$Loss_D = \frac{1}{2} \mathbb{E}_{X \sim P_{data}(X)} [D(E(X), Y) - 1]^2 + \frac{1}{2} \mathbb{E}_{z \sim P_z(z)} [D(G(z, Y), R(E(D_z(G(z, Y)))))]^2 \quad (7)$$

where  $E$  is the encoder and  $E(X)$  is the latent vector output from the encoder. In the pretraining process when the reinforcement has not been activated, the subscript  $X \sim P_{data}(X)$  indicates that the molecule is sampled from the training data, and  $z \sim P_z(z)$  refers to all synthesized samples generated by the generator. After the reinforcement is activated,  $X \sim P_{data}(X)$  refers to the generated samples that are chemically valid and have the predicted  $C_v$  with RE of < 20%, and  $z \sim P_z(z)$  refers to the generated samples that do not pass either of the validity or accuracy tests.

**Training of RRCGAN.** The process starts with training AE and the regressor using ~63K molecules from the QM9 library.<sup>29</sup> AE was trained by minimizing the discrepancy between the input SMILES to the encoder and the output ones from the decoder. Fig. S7 shows that the loss of AE is stabilized after 300 epochs. Comparison of the true one-hot encoded SMILES strings with those outputted from the decoder shows that the decoder can accurately convert the continuous vectors back to their discrete molecule representation (Fig. S8). The latent vectors outputted from the pre-trained encoder and the corresponding  $C_v$  of these molecules are used to train the regressor. Fig. S9 shows that the loss of the regressor is stabilized after 75 epochs. As shown in Fig. S10, the resulting regressor affords the coefficient of determination, R-squared ( $R^2$ ) of 0.99, and a root mean-square-error (RMSE) of 0.34 cal/(mol·K) for training and  $R^2$  of 0.98 and RMSE of 0.42

cal/(mol·K) for testing, respectively, indicating the high model accuracy. This pre-trained regressor is used to predict  $C_v$  of the synthesized molecules from the generator. It is also used in the reinforcement center to screen the samples with unsatisfactory  $C_v$  accuracy.

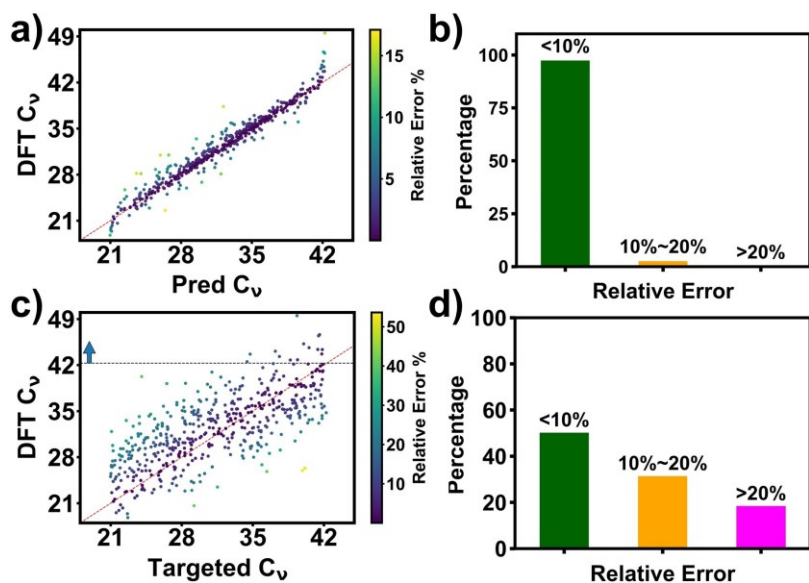
After pretraining AE and the regressor, the generator and discriminator are first trained for 5 epochs. After five epochs, the reinforcement is activated. After the reinforcement is active, the generator generates 1000 latent vectors in response to the input targeted  $C_v$ . The reinforcement center groups the molecules based on two criteria: the SMILES validity and accuracy of the predicted  $C_v$ . To check the validity of the generated molecules, their latent vectors are first converted to SMILES by the decoder and then validated by RDKit. Meanwhile, these SMILES are converted to the new latent features and then fed to the pre-trained regressor for predicting  $C_v$ . The reinforcement center selects the generated samples that are chemically valid and have the predicted  $C_v$  within RE of 20% of the targeted values. These selected samples are labeled as “1” and the remaining ones are labeled as “0”. Then, these grouped samples are fed to train the discriminator. The loss function of the discriminator favors towards distinguishing the real and fake samples. The evolution loss of the generator and discriminator is represented in Fig. S11. It shows that after the reinforcement center is activated, the loss of the generator is fast reduced and stabilized after 100 epochs. The discriminator’s loss increases very little until the 75<sup>th</sup> epoch and is stabilized thereafter. The low and stabilized losses of both the generator and discriminator indicate a successful model training. Hyperparameters for these trained networks are shown in Table S1. Evaluation metrics such as  $R^2$ , mean absolute error (MAE), RMSE, MSE, and RE are defined in Eq. S1-Eq. S5 (Supplementary Note 3).

## 2.2 Evaluation of RRCGAN

The performance of RRCGAN was evaluated by comparing the calculated  $C_v$  of the generated molecules with the targeted  $C_v$  and the predicted  $C_v$  by the regressor, respectively. The initial samples that were used to train the model were in the range of 20.9 and 42.3 cal/(mol·K), and the  $C_v$  values were normalized to continuous values between 0 and 1 using 20.9 as minimum and 42.3 as maximum. ~500 unique, novel generated molecules were calculated by density functional theory (DFT). The predicted  $C_v$  values by the regressor were first compared with the DFT calculated ones (Fig. 2a). Their  $R^2$  and RMSE were calculated to be 0.94 and 1.2 cal/(mol·K), respectively. This high prediction accuracy suggests that the well-trained regressor catches the hidden chemical rules to correlate the structures of molecules with their properties. It should be noted that the accuracy deteriorates for the samples with very low or very high  $C_v$ . That's because the minimum and maximum  $C_v$  of the training data are 20.9 and 42.3 cal/(mol·K), respectively. The regressor can only predict them accurately in this range. Fig. 2b shows the RE distribution of the predicted  $C_v$  by the regressor compared with the DFT calculated ones. ~98% of the molecules show  $\leq 10\%$  RE of the DFT-calculated values. The results shown in Fig. 2a-b suggest a high accuracy of the regressor in predicting  $C_v$  of the generated molecules with the DFT calculation. Thus, it is acceptable to directly use the regressor to screen the generated molecules, which would save time and cost from using the DFT calculation.

In addition, the targeted  $C_v$  and DFT-evaluated  $C_v$  of the generated molecules were compared to evaluate the accuracy of the RRCGAN model in generating novel molecules (Fig. 2c). The data shows  $R^2$  and RMSE of 0.50 and 4.6 cal/(mol·K), respectively. Distribution of RE between the DFT-calculated and targeted  $C_v$  is shown in Fig. 2d. ~80% of the molecules have  $C_v$  calculated by DFT within 20% RE of the targeted values, showing an acceptable accuracy in such *de novo* generation. Among them, 2.8% of all the generated molecules have  $C_v$  of  $> 42.3$  cal/(mol·K) with

a maximum value of 49.5 cal/(mol·K). Some of these outlier molecules are summarized in Fig. S12. As a comparison, RRCGAN was also trained with another type of string-based molecule representation, named SELFIES (Supplementary Note 1).<sup>34</sup> As shown in Fig. S13, the model accuracy is far worse than that of the model trained with SMILES. The reason is that the algorithm converts the invalid strings to valid ones, while the generator does not learn how to generate valid SELFIES. An obvious disadvantage of any string-based representation method, e.g., SMILES, is that information about bond lengths and 3D configurations is lost. Trained with molecules presented by them, the model shows a limitation in the accuracy. Better accuracy may require more input information like the molecules' 3D configuration, while it is a trade-off between the amount of the input information and the computational cost. In future, we plan to embed some 3D information using a distance geometry methods<sup>35</sup> to the SMILES vectors to validate whether the accuracy of RRCGAN can be improved.

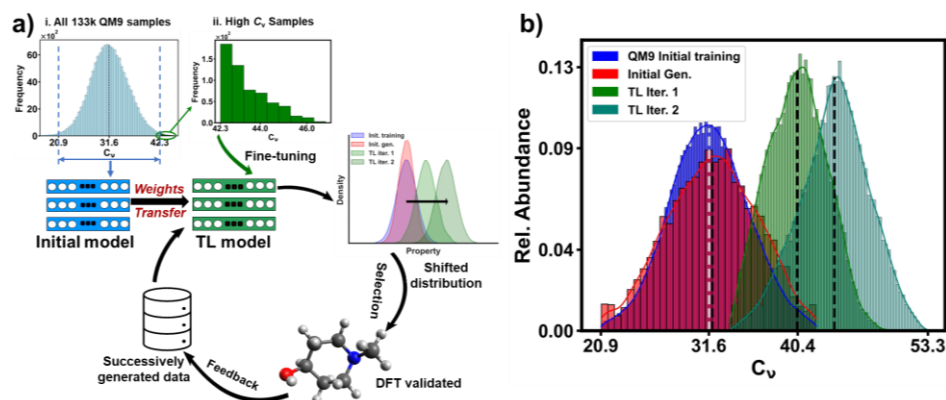


**Figure 2. Comparison of targeted, predicted, and DFT calculated  $C_v$  for the generated molecules.** Absolute value distribution (a) and RE distribution (b) of DFT calculated and regressor predicted  $C_v$ . Absolute value distribution (c) and RE distribution (d) of DFT calculated and targeted  $C_v$ .

### 2.3 Transfer learning for biasing $C_v$ towards higher values

The  $C_v$  statistical results of all 133K QM9 are shown in Table S2, and they follow a normal distribution (Fig. 3a-i). To train the original RRCGAN model, we used the  $1.5 \times \text{IQR}$  rule to exclude outlier molecules with  $C_v$  of  $< 20.9$  and  $> 42.3$  cal/(mol·K) from the original samples. Because these outliers are only 1% of the total samples, including them in the model training will significantly degrade the accuracy of the model in generating the molecules with high  $C_v$  due to the lack of data. After removing outliers, we randomly sampled ~63K samples with  $C_v$  statistical results shown in Table S3 to train the initial RRCGAN. The 63K training samples follow the same distribution as that of the total QM9 samples (Fig. 3a-i). Although the originally trained RRCGAN model may generate some outlier molecules with DFT-calculated  $C_v$  of  $> 42.3$  cal/(mol·K) (Fig. S12), the number is very low (only 2.8% of the batch). In contrast, transfer learning has shown a great promise in solving problems in material discovery when the data is limited.<sup>36</sup> It works by transferring the knowledge from an already trained model to a new one, thus improving the accuracy of the new model even if trained with limited data.<sup>37</sup> Thus, to bias the  $C_v$  towards higher values for extrapolating the property space, a transfer learning model was trained via finetuning the initially trained RRCGAN model based on ~2000 molecules with  $C_v$  of  $> 42.3$  cal/(mol·K). 587 of them were the outliers in the QM9 library (Fig. 3a-ii). The rest were the newly generated outlier molecules. We first screened the ones with  $C_v$  of  $> 40$  cal/(mol·K) predicted by the regressor and then validated them by the DFT calculation. Then, the DFT calculated  $C_v$  values were labeled

as part of a new training dataset to retrain the transfer learning model. The workflow of such an iterative searching algorithm is shown in Fig. 3a. As a demo, herein, only two iterations were investigated. In the second iteration, the transfer learning model was fine-tuned by the generated molecules by the first iterated model, which show validated  $C_v$  of  $> 42$  cal/(mol·K). Fig. 3b shows the distributions of  $\sim 63$ K initial samples used for training the original RRCGAN model and the initially generated samples from it. Predicted  $C_v$  of the generated samples is in the range of 20.9 and 42.2 cal/(mol·K) with the average of 31.9 cal/(mol·K), which is close to that of the training samples, 31.6 cal/(mol·K). After the first iteration, the transfer learning model generates the molecules with the mean  $C_v$  of 40.4 and a maximum value of 50.7 cal/(mol·K). The percentage of the molecules with  $C_v$  of  $> 42.3$  cal/(mol·K) is 26%. After the 2<sup>nd</sup> iteration, it is found that the generated molecules have a mean  $C_v$  of 44.0 cal/(mol·K) and a maximum  $C_v$  value of 53.3 cal/(mol·K). The percentage of the molecules with  $C_v$  of  $> 42.3$  cal/(mol·K) is 71%. The results shown in Fig. 3b illustrate that the iterative transfer learning can improve the percentage of generated molecules with the iteratively increased  $C_v$ .



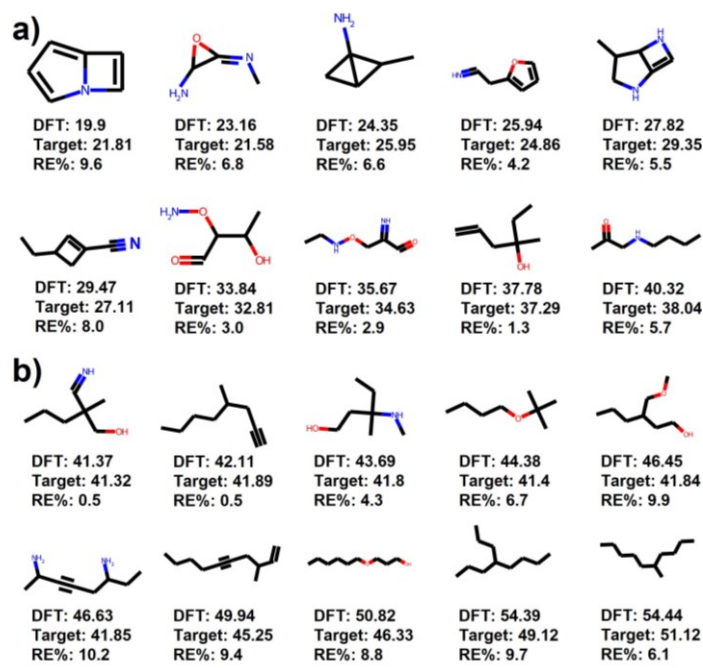
**Figure 3.** Workflow of the iterative transfer learning and the model performance. (a) Schematic of transfer learning in the process of going beyond the range of initial training data by iteratively

learning from the generated data. **(b)**  $C_v$  distributions of the initial training molecules in the QM9 library, the molecules generated by the original model, and the molecules generated by the 1<sup>st</sup> transfer learning model and the 2<sup>nd</sup> transfer learning model.

## 2.4 Analysis of the generated molecules

**Analysis of structural features of molecules.** The aforementioned active search strategy makes it possible to generate novel molecules with  $C_v$  of higher than the ones in the original training dataset. It can be deduced that RRCGAN has successfully learned the chemical rules contained in SMILES to establish the structure-property relationship. Herein, we analyzed the structures of the generated molecules to understand how the model can capture chemical insights. 20 representative molecules whose DFT-calculated  $C_v$  are in the range of 19.9-54.44 cal/(mol·K) and have RE within  $\leq 10\%$  of the targeted values are shown in Fig. 4. Fig. 4a shows the ones with  $C_v$  of  $< 41$  cal/(mol·K), while Fig. 4b shows the ones with  $C_v$  of  $> 41$  cal/(mol·K). The comparison shows that the molecules having the ring structures, N atoms (especially the ones with double bonds), O atoms, and branches exhibit lower  $C_v$  values, while those having the long linear carbon chains and fewer branches tend to have higher  $C_v$  values. This observation validates that the model does correlate the patterns of the structures with the property values, which agrees well with the established chemical rules.<sup>38</sup> This good agreement improves the trust in using the deep generative model to generate novel molecules with targeted properties in a time- and cost- effective manner.

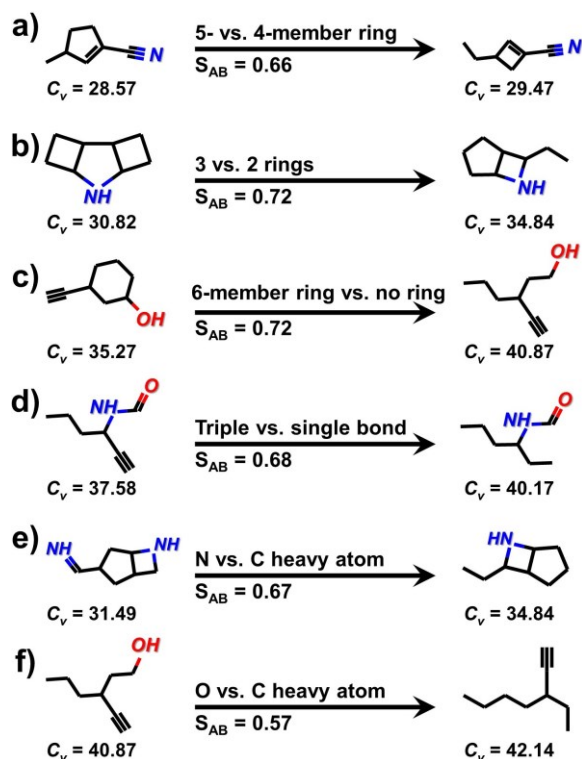




**Figure 4. Representative examples of molecules generated by the original and transferred RRCGAN models: the molecules with (a)  $C_v$  of  $< 41$  cal/(mol·K), and (b)  $C_v$  of  $> 41$  cal/(mol·K).**

To further investigate how the structural features of molecules affect their  $C_v$ , we performed a one-to-one comparison between the ones with a high structural similarity (Fig. 5). To do that, first, we used RDKit to extract bit vectors (fingerprints) that represent the molecules. From those fingerprints, we calculated their Tanimoto coefficient ( $S_{AB}$ )<sup>39</sup> (Supplementary Note 4, Equation S6). Molecules with a close  $S_{AB}$  have quantitatively similar structures. We focused on the molecules with the close  $S_{AB}$  but different  $C_v$ . Fig. 5a represents two molecules with  $S_{AB}$  of 0.66 and the same number of heavy atoms (HAs = 8). The molecule with a 5-member ring (left) has a smaller  $C_v$  than the one with a 4-member ring (right), suggesting that more atoms in a ring decrease

$C_v$ . What's more, it is hypothesized that the increased number of rings in a molecule will lead to a decrease in  $C_v$ . This hypothesis is well supported by the result shown in Fig. 5b-c. Fig. 5b represents two molecules with  $S_{AB}=0.72$  and the same HA number (HAs = 9). The molecule with 3 rings (left) shows a lower  $C_v$  value than the one with 2 rings (right), while the molecule without any rings has the highest  $C_v$  value (Fig. 5c). We also studied the effect of carbon bond number. Fig. 5d shows that the molecule with a triple bond (left) has a smaller  $C_v$  than the one with a single bond, indicating that presence of a triple bond may lower the  $C_v$  value. In addition, the impact of some specific functional groups in  $C_v$  was studied. As shown in Fig. 5e, replacing the C=NH group with the C-C group in the same structure increases the  $C_v$  value, while the molecule with the presence of the -OH group shows a reduced  $C_v$  value (Fig. 5f). In summary, after analyzing the features present in the generated molecules, we found that the presence of the rings, triple and double bonds, and some types of atoms such as nitrogen and oxygen would decrease the  $C_v$  value. This conclusion is in good agreement with the established group contribution methods, which help to predict the specific heat of some molecules. While they cannot always correlate the structures with the properties, it is reasonable to conclude that the proposed RRCGAN model would afford a trustworthy, purely data-driven methodology for the highly efficient generation of novel lead compounds without any physical or chemical inputs.

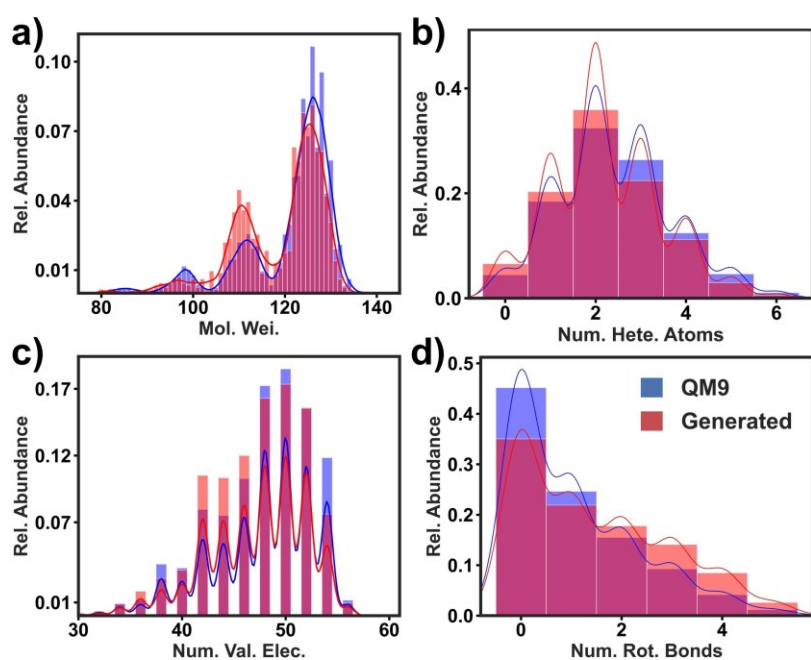


**Figure 5.** Effect of structural features in the generated molecules on their  $C_v$ . (a) Structures of the molecules having the same HA number but different ring size. (b, c) Structures of the molecules having 3, 2, 1, 0 rings, respectively. (d) Structures of the molecules with triple and single bonds. (e-f) Structures of the molecules with different nitrogen (e) and oxygen (f) atoms.

Visualizing these representative molecules in **Fig. 5** affords a qualitative correlation of the structures with their  $C_v$ . To make such correlation quantitative, we trained an XGBoost regression model which takes 18 structural features as input to predict the output  $C_v$ . These features include molecular weight (MW), number of heavy atoms, MW of heavy atoms, number of hydrogen acceptors, number of Hydrogen donors, number of hetero atoms, number of rotatable bonds, the

sum of valence electrons in heavy atoms, number of aromatic rings, number of saturated rings, number of aliphatic rings, number of radical electrons, number of aliphatic carbocycles, number of aliphatic heterocycles, number of aromatic carbocycles, number of aromatic heterocycles, number of saturated carbocycles, and number of saturated heterocycles. The hyperparameters of the well-trained model are shown in Table S4. The prediction  $R^2$  of the XGBoost regression model is 0.91 for training and 0.90 for testing samples. Fig. S14 shows the rankings of the input features in determining the  $C_v$  values. The result shows that the sum of valence electrons is the most important feature. Following that, the number of hetero atoms, number of rotatable bonds, number of aliphatic rings, and MW are ranked as the subsequently important features. This observation agrees well with the results shown in **Fig. 4**. To further investigate the feature importance in determining  $C_v$ , the four features: molecular weight, number of heteroatoms, the sum of valence electrons, and number of rotatable bonds were selected. **Fig. 6** shows the density distribution of these selected features presented in both the training and generated molecules. The feature distributions of the generated molecules with  $C_v$  in the range of 20.9-42.3 cal/(mol K) follow those of the training ones, showing the generator's ability to generate the molecules with the targeted  $C_v$  by appropriately weighting these features. Specifically, as shown in Fig. 6a-b, most of the training and generated molecules have MW in the range of 120 and 135 and 1-3 heteroatoms. Also, the sum of valence electrons of the heavy atoms for most molecules is ~50 (Fig. 6c). Fig. 6d shows that most molecules have zero rotatable bonds and the ratio of the generated molecules with zero rotatable bonds is slightly lower than that of training samples. It indicates that the number of rotatable bonds is shifted toward a higher value for the generated molecules. These average values of the features are related to the molecules with  $C_v$  of ~31 cal/(mol·K), which is consistent with the training molecules with an average  $C_v$  of 31.6 cal/(mol·K). In addition to  $C_v$ , we also

investigated three drug quality metrics—octanol-water coefficient (log P) quantitative estimation of drug-likeness (QED), and topological polar surface area (TPSA)—which are considered important for the drug discovery.<sup>2</sup> Fig. S15 shows that the distribution of them for generated molecules follow well with that for the training ones. Details about them are explained in Supplementary Note 5.

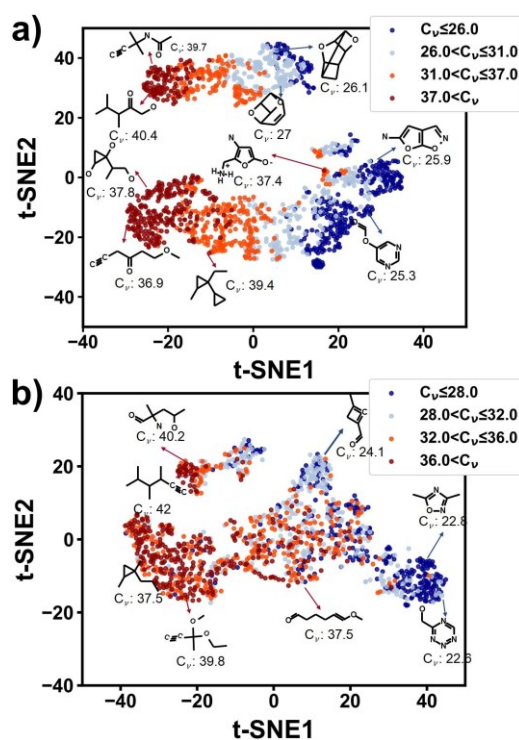


**Figure 6. Density distribution of the four selected features for the generated versus training samples: (a) molecular weight; (b) number of heteroatoms; (c) number of valence electrons; (d) number of rotatable bonds.**

**Dimension reduction on the latent vectors of the molecules.** The latent vectors, which are the output of the encoder, were used as the input for training the RRCGAN model. These

continuous vectors can connect the discrete SMILES representations with their hidden structural information for generating novel molecules with their targeted properties. They are, however, high-dimensional and are difficult to be interpreted. We hypothesize that if they can be mapped to the low-dimensional space, the molecules that share similar structural features would be clustered together in the plots, and the generated molecules would follow a similar pattern to the training ones. To validate the hypotheses, we applied t-distributed stochastic neighbor embedding (t-SNE), a non-linear dimension reduction method, to project the latent vectors of both training and generated molecules (Fig. 7). First, we divided  $C_v$  into four ranges. Each range was calculated based on quantile in a way that has the same number of molecules in each range. The projected latent vectors were then colored based on their  $C_v$  ranges, where the dark blue and dark red colors represent the low and high values, respectively. As shown in Fig. 7, the values of the first component of t-SNE (t-SNE1) separate the samples based on their  $C_v$  values. Samples with higher  $C_v$  are in a region with low t-SNE1 and vice versa. For both the training and generated samples, most blue molecules with t-SNE1 > 0 have  $C_v$  of < 32 cal/(mol·K). The values of the second component of t-SNE (t-SNE2), on the other hand, cannot separate the samples based on their  $C_v$  values. The molecules in the same  $C_v$  range are clustered into close regions in the plots. The linear molecules with fewer heteroatoms are grouped in the high  $C_v$  range, while the molecules with rings occupy the low  $C_v$  regions. This result is consistent with our observations from Fig. 4-5. Also, there are two regions with low and high  $C_v$  values in the training samples (Fig. 7a). These two regions are separated based on the second component magnitude, t-SNE2. By looking at the structures of the low  $C_v$  samples, one can see that the 4-membered rings are dominated in the molecules with t-SNE2 > 20, while 5- and 6-membered rings are dominated in the molecules with t-SNE2 < 20 (Fig. 7a). The same separation happens to the generated samples as shown in Fig. 7b, but there are

fewer molecules in the upper region. Moreover, the generated molecules are clustered in the same regions as the ones for the training molecules (Fig. 7b), further validating that the generator has successfully learned the structural information from the latent space of the training molecules for generating novel molecules with biased and targeted  $C_v$ . As a comparison, we also performed a principal component analysis (PCA) and a spectral embedding analysis on the same molecules used for the t-SNE analysis. The results are shown in Fig. S16. Discussion on the PCA and spectral embedding results is described in Supplementary Note 6. In conclusion, it is found that t-SNE outperforms the other two methods for data visualization in this case.



**Figure 7. t-SNE plots of latent features of the training and generated molecules output from the encoder: (a) training molecules; (b) generated molecules.**

### 3. Conclusion

In this work, we designed and implemented a deep generative framework named RRCGAN for *de novo* design of molecules with targeted properties. To develop the model, we first trained the encoder and decoder. Then, the encoded latent features of the molecules are fed to the regressor to predict  $C_v$ , which helps GAN to generate targeted and valid compounds. It is worth mentioning that only *SMILES* is used as the input of the model, and no other complicated chemical descriptors are employed in the study.  $C_v$  values of the generated molecules are calculated by DFT for validation and comparison with the targeted values. The developed RRCGAN is transferred by using the limited, generated molecules in the previous iteration for the next-iteration generation of new molecules to bias  $C_v$  toward the values beyond the initial training data.

To the best of our knowledge, this is one of the first attempts at developing a combined DL and iterative transfer learning strategy for *de novo* design of novel molecules with the targeted property values beyond the initial training data. After only two transfer learning iterations, the generated molecules show the increased mean  $C_v$  of 44.0 cal/(mol·K) compared to 31.6 cal/(mol·K) in the original QM9 library. To validate the effectiveness and trustworthiness of the model, the structures and the latent features of both the training and generated molecules are qualitatively and quantitatively analyzed. Through the analysis, it is disclosed that the model has successfully learned the hidden structural features and correlated them with the properties for generating novel molecules with biased and targeted  $C_v$ , which agrees well with the established physical and chemical rules. The proposed RRCGAN framework would afford a trustworthy, purely data-driven methodology for the highly efficient generation of novel lead compounds or other matter without physical or chemical inputs.



## 4. Methods

**Data collection and curation.** After removing the outliers from ~133K samples in the QM9 library,<sup>29</sup> we randomly sampled ~63K molecules for training the original RRCGAN model. Canonical *SMILES* were used to represent the molecules.<sup>40</sup> To one-hot encode *SMILES*, a subset of 23 different characters was used. All the training molecules have less than or equal to 9 heavy atoms of C, O, N, and F. The maximum sequence length is 35 characters. Moreover, all the sequences were padded to the maximum sequence length. These *SMILES* representations were split into training, validation, and test datasets in a ratio of 6:2:2. The training and validation datasets were used to finetune the hyperparameters of the encoder, decoder, and regressor, while the test datasets were used to evaluate the final performance of the model.

**DFT calculation.** We used Avogadro 1.2.0, an open-source molecular builder and visualization tool,<sup>41</sup> to convert the generated 2D strings to 3D coordinates. We employed an auto-optimization tool from Avogadro with a universal force field (UFF) to optimize the molecular geometries. The optimized molecules were then fed into Gaussian 16 for further geometry optimization and then calculating the heat capacity at ambient temperature.<sup>42</sup> Heat capacities at ambient temperature were calculated at the B3LYP level of quantum chemistry from the 6-31G(2df,p) basis set.<sup>43</sup> This method is the basis for the *Gn* methods, which have an accuracy close to the experimental one.<sup>29,44</sup> Gaussian software uses four components to calculate the heat capacity, namely electronic, translational, rotational, and vibrational motions.<sup>45</sup> Partition function,  $Q(N, V, T)$ , where  $N$  is the total number of particles,  $V$  is the volume, and  $T$  is the temperature of the system, was used for each component under the thermodynamic equilibrium.<sup>45</sup> Since there is no temperature-dependent term in the partition function, the heat capacity in the constant volume

from electronic motion is zero.<sup>45</sup> Gaussian assumes non-interacting particles, an assumption that is applied for only ideal gas. Moreover, for the electronic calculations, Gaussian assumes that the first and higher excited states are inaccessible.<sup>44</sup> To ensure that the calculated values are close to those in the QM9 library, we calculated the heat capacities of 47 samples randomly selected from the QM9 library. The calculated values were compared with the ones listed in the QM9 library. The result shows that they have a low MAE of 0.12 cal/(mol·K) (Fig. S17).

#### **Data and code availability**

The corresponding data and codes can be available at <https://github.com/linresearchgroup/XX>.

#### **Acknowledgements**

J. L thanks financial support by National Science Foundation (award number: 2154428) and U.S. Army Corps of Engineers, ERDC (grant number: W912HZ-21-2-0050). O. I. acknowledges the support from National Science Foundation (award number: 2154447). The computation for this work was performed on the high-performance computing infrastructure provided by Research Computing Support Services at the University of Missouri, Columbia MO, which is in part supported by National Science Foundation (Award number: CNS-1429294).

#### **Contributions**

J.L conceived the idea. D.L. developed the original framework which was significantly modified by K.S.. K.S. performed the model training and data analysis. K.S. and J.L. wrote the complete manuscript. Y.X. assisted K.S. in writing the first draft. O.I. provided valuable discussion and contributed the writing about the analysis of the generated molecules. J.L. oversaw all research

phases and provided guidance to the research team. All authors discussed and commented on the manuscript.

### Additional information

Supplementary information is available.

### Competing interests

The authors declare no competing financial interests.

### References

- 1 Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: A molecular modeling perspective. *Med. Res. Rev.* **16**, 3-50 (1996).
- 2 Polishchuk, P. G., Madzhidov, T. I. & Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput. Aided Mol. Des.* **27**, 675-679 (2013).
- 3 Gómez-Bombarelli, R. *et al.* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **4**, 268-276 (2018).
- 4 Gómez-Bombarelli, R. *et al.* Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat. Mater.* **15**, 1120-1127 (2016).
- 5 Reymond, J.-L. The Chemical Space Project. *Acc. Chem. Res.* **48**, 722-730 (2015).
- 6 Hachmann, J. *et al.* The Harvard Clean Energy Project: Large-Scale Computational Screening and Design of Organic Photovoltaics on the World Community Grid. *J. Phys. Chem. Lett.* **2**, 2241-2251 (2011).
- 7 Carrete, J., Li, W., Mingo, N., Wang, S. & Curtarolo, S. Finding Unprecedentedly Low-Thermal-Conductivity Half-Heusler Semiconductors via High-Throughput Materials Modeling. *Phys. Rev. X* **4**, 011019 (2014).
- 8 Pyzer-Knapp, E. O., Suh, C., Gómez-Bombarelli, R., Aguilera-Iparraguirre, J. & Aspuru-Guzik, A. What Is High-Throughput Virtual Screening? A Perspective from Organic Materials Discovery. *Annu. Rev. Mater. Res.* **45**, 195-216 (2015).
- 9 Rupakheti, C., Virshup, A., Yang, W. & Beratan, D. N. Strategy To Discover Diverse Optimal Molecules in the Small Molecule Universe. *J. Chem. Inf. Model.* **55**, 529-537 (2015).
- 10 Douguet, D., Thoreau, E. & Grassy, G. A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm. *J. Comput. Aided Mol. Des.* **14**, 449-466 (2000).
- 11 Dong, Y. *et al.* Inverse design of two-dimensional graphene/h-BN hybrids by a regressional and conditional GAN. *Carbon* **169**, 9-16 (2020).
- 12 Sattari, K., Xie, Y. & Lin, J. Data-driven algorithms for inverse design of polymers. *Soft Matter* **17**, 7607-7622 (2021).

- 13 Sahu, H. *et al.* Designing promising molecules for organic solar cells via machine learning assisted virtual screening. *J. Mater. Chem. A* **7**, 17480-17488 (2019).
- 14 Li, X. *et al.* Combining machine learning and high-throughput experimentation to discover photocatalytically active organic molecules. *Chem. Sci.* **12**, 10742-10754 (2021).
- 15 Tiihonen, A. *et al.* Predicting Antimicrobial Activity of Conjugated Oligoelectrolyte Molecules via Machine Learning. *J. Am. Chem. Soc.* **143**, 18917-18931 (2021).
- 16 Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. Preprint at <https://ui.adsabs.harvard.edu/abs/2013arXiv1312.6114K> (2013).
- 17 Goodfellow, I. *et al.* Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27** (2014).
- 18 Mnih, V. *et al.* Playing atari with deep reinforcement learning. Preprint at <https://arxiv.org/abs/1312.5602> (2013).
- 19 Narasimhan, K., Kulkarni, T. & Barzilay, R. Language understanding for text-based games using deep reinforcement learning. Preprint at <https://arxiv.org/abs/1506.08941> (2015).
- 20 Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **4**, 120-131 (2018).
- 21 Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **4**, eaap7885 (2018).
- 22 Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **361**, 360-365 (2018).
- 23 Elton, D. C., Boukouvalas, Z., Fuge, M. D. & Chung, P. W. Deep learning for molecular design-a review of the state of the art. *Mol. Syst. Des. Eng.* **4**, 828-849 (2019).
- 24 Kadurin, A. *et al.* The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* **8**, 10883 (2017).
- 25 Xie, Y., Sattari, K., Zhang, C. & Lin, J. Toward autonomous laboratories: Convergence of artificial intelligence and experimental automation. *Prog. Mater. Sci.* **132**, 101043 (2023).
- 26 Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C. & Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. Preprint at <https://arxiv.org/abs/1705.10843> (2017).
- 27 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). Preprint at <https://doi.org/10.26434/chemrxiv.5309668.v3> (2017).
- 28 Cohen, N. & Benson, S. W. Estimation of heats of formation of organic compounds by additivity methods. *Chem. Rev.* **93**, 2419-2438 (1993).
- 29 Ramakrishnan, R., Dral, P. O., Rupp, M. & von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **1**, 140022 (2014).
- 30 Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31-36 (1988).
- 31 Szegedy, C. *et al.* Going deeper with convolutions. In Proc. *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. 1-9 (2015).
- 32 Landrum, G. *Open-source Cheminformatics Software*, (2006); <https://www.rdkit.org/>
- 33 Mao, X. *et al.* Least Squares Generative Adversarial Networks. Preprint at <https://arxiv.org/abs/1611.04076> (2016).

- 34 Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **1**, 045024 (2020).
- 35 O'Boyle, N. M. *et al.* Open Babel: An open chemical toolbox. *J. Cheminf.* **3**, 33 (2011).
- 36 Zubatyuk, R., Smith, J. S., Leszczynski, J. & Isayev, O. Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network. *Sci. Adv.* **5**, eaav6490 (2019).
- 37 Yuan, R. *et al.* Accelerated Discovery of Large Electrostrains in BaTiO<sub>3</sub>-Based Piezoelectrics Using Active Learning. *Adv. Mater.* **30**, 1702884 (2018).
- 38 Wunderlich, B. *Thermal Analysis of Polymeric Materials*. (Springer Science & Business Media, Berlin, Heidelberg, 2005).
- 39 Bajusz, D., Rácz, A. & Héberger, K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminf.* **7**, 20 (2015).
- 40 O'Boyle, N. M. Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *J. Cheminf.* **4**, 22 (2012).
- 41 Hanwell, M. D. *et al.* Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *J. Cheminf.* **4**, 17 (2012).
- 42 Frisch, M. J. *et al.* Gaussian 16 Rev. C.01 (Gaussian, Inc., Wallingford CT, 2016).
- 43 Tirado-Rives, J. & Jorgensen, W. L. Performance of B3LYP Density Functional Methods for a Large Set of Organic Molecules. *J. Chem. Theory Comput.* **4**, 297-306 (2008).
- 44 Curtiss, L. A., Redfern, P. C. & Raghavachari, K. Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* **127**, 124105 (2007).
- 45 McQuarrie, D. A. & Simon, J. D. *Physical chemistry: a molecular approach*. Vol. 1 (University Science Books, Sausalito, CA, 1997).

## Supporting Information for

### ***De Novo* Design of Molecules Towards Biased Properties via a Deep Generative Framework and Iterative Transfer Learning**

Kianoosh Sattari<sup>1</sup>, Dawei Li<sup>1</sup>, Yunchao Xie<sup>1</sup>, Olexandr Isayev<sup>4</sup>, and Jian Lin<sup>1, 2, 3\*</sup>

<sup>1</sup>Department of Mechanical and Aerospace Engineering,

<sup>2</sup>Department of Electrical Engineering and Computer Science

<sup>3</sup>Department of Physics and Astronomy

University of Missouri, Columbia, Missouri 65211, United States

<sup>4</sup>Department of Chemistry, Carnegie Mellon University, Pittsburgh, PA 15213, United States

\*E-mail: LinJian@missouri.edu (J. L.)

## Supplementary Notes

### **Supplementary Note 1: Molecule representations by SMILES versus SELFIES**

Molecules can be represented as undirected graphs.<sup>1</sup> Each atom is considered as a node, and bonds are considered edges connecting the nodes. Introduced by David Weininger in 1988,<sup>2</sup> *SMILES* (Simplified Molecular Input Line Entry System) have been used as a representation approach in computational chemistry and cheminformatics.<sup>3</sup> *SMILES* is a string-based representation method. It is based on a molecular graph theory that defines molecular structures with predefined grammar rules. By following the specific rules, *SMILES* represents the topology of a molecule as a standard molecular graph.<sup>4</sup> It is true that *SMILES* only includes 2D molecular information, but an accurate prediction in the properties of molecules in equilibrium do not need all conformational degrees of freedom of the molecules.<sup>5-7</sup> In the *SMILES* representation, atomic symbols represent heavy atoms (e.g. C, N, F, and O), “=” and “#” represent bond types (double and triple, respectively), numbers represent rings, and parentheses represent branches within a molecular structure.<sup>2</sup> To reduce the complexity, hydrogen atoms are removed since they can be deduced from the chemistry valence rules.<sup>8</sup>

There are two sources of non-uniqueness in *SMILES*. First, there is an ambiguity about which atom to start the *SMILES*. Second, the choice of whether to include charge information in the

resonance structure makes the *SMILES* representations not unique. The canonical *SMILES* following the standard rules defines the atoms and bonds of molecules in a defined order. Thus, there is a unique canonical *SMILES*. It should be noted that the canonical *SMILES* is considered neither a universal nor a global identifier since there are various code rules for generating *SMILES* in each system. We used an open-source cheminformatics suite, RDKit<sup>9</sup> for both the input dataset and the generated samples. It shows a unique *SMILES* for a given molecule.

The major challenge of using the *SMILES* representation is that a large fraction of strings generated by a probabilistic model do not represent valid molecules.<sup>3</sup> The generated sequences are either syntactically invalid where the strings do not follow the *SMILES* grammar rule, or semantically invalid where they do not follow the chemistry rule. For instance, the sequence “CCCC(OCC” has an open parenthesis but not a closed one. The sequence “c1cccc” has a starting point for a ring but no a closing point. They are examples of semantically invalid sequences. The string “CO(C)C”, on the other hand, does not follow the chemistry rule since number of the explicit valence state for oxygen is greater than the one permitted. It is semantically invalid.

Researchers have proposed a modified *SMILES* to solve the mentioned validity problems. They include *DeepSMILES*<sup>10</sup> and *SELFIES*.<sup>3</sup> In the *SMILES* syntax, branches are represented by balanced pairs of parentheses, an open parenthesis followed by atoms inside the branch and a close parenthesis to end the branch. For example, Isobutyric acid with two branches is represented as “CC(C)C(=O)O”. Moreover, rings are indicated by a pair of digits with the atoms between the two digits. For example, “c1cccc1” represents a benzene ring consisting of 6 carbons. Thus, for the rings and branches, *SMILES* uses two symbols that must occur in pairs. *DeepSMILES*, invented by O’Boyle and Dalke,<sup>10</sup> solves the unbalanced parentheses problem in *SMILES*. It defines the branches by only the close parentheses. Also, it uses only one digit to show the ring. In future, we

plan to investigate *DeepSMILES*. Krenn et al. very recently solved the problem at a fundamental level by introducing *SELFIES* (SELF-referencing Embedded Strings). It is a string-based representation method like *SMILES* but has 100% robustness.<sup>3</sup> In *SELFIES*, all the combinations of the strings are valid. However, the derivation rules are complicated and take lots of effort to be designed for a specific dataset. In this work, we tried *SELFIES* and got 100 valid generated strings. However, the accuracy of the generated samples is not comparable to that of the generated samples represented by *SMILES* (Figure S13). The reason is that the derivation rules defined in *SELFIES* force the strings to be valid even if the generated samples may not be valid. Since the regressor is only trained on the latent vectors of valid strings, it cannot accurately predict the property of an invalid latent vector. For example, if the generator generates a “[Branch1][Branch2][F][=C][=C][#N]” invalid sequence, the derivation rules convert it to “[F][=C][=C][#N]”. In this way, the generator does not receive any penalty for generating such an invalid sequence. Also, the regressor should predict the property from the former one. For more details on the derivations rules and syntaxes that are used, one is suggested to refer to the paper.<sup>3</sup>

## **Supplementary Note 2: Design and training of the reinforced regressional and conditional GAN (RRCGAN)**

**Architecture of the encoder.** The encoder is a convolutional neural network (CNN) with an architecture shown in Fig. S2. It outputs fixed-dimensional latent vectors ( $6 \times 6 \times 2$ ) that have the most statistically important information from the input discrete one-hot encoded matrices. As shown in Fig. S2, the encoder has two parallel networks that were fed with the same one-hot encoded matrices. Each layer has 4 sequential convolutional blocks that gradually reduce the size of the output of the previous layers that start from  $36 \times 23 \times 1$  and end with  $6 \times 6 \times 1$ . Each



convolutional block consists of a convolutional layer, a leaky ReLU activation function (AF), and a batch normalization layer. The four convolutional blocks are followed by a convolutional layer and a Tanh AF to output a  $6 \times 6 \times 1$  vector with continuous number between -1 and 1. The two output of each parallel networks are then concatenated, resulting in the final latent matrices with  $6 \times 6 \times 2$  dimensions. We hypothesized that one of the two parallel networks in the encoder architecture relates to atoms information and the another relates to bonds information.

**Architecture of the decoder.** The architecture of the decoder was modified from Google Inception V2, shown in Fig. S3. The decoder converts the latent vectors back to the original SMILES strings (input to the encoder). The advanced architecture of Inception V2 allows for increasing the depth and width of the network to convert the continuous vectors back to original discrete SMILES representation. The original Inception V2 model has been used for classification tasks. Here, the decoder is used for a similar task, as it should come up with a probability for each possible 23 characters for every 35 different positions.

**Architecture of the regressor.** The structure of the regressor is shown in Fig. S4. It was modified from the Google Inception V2 model. The inception modules are activated by a leaky rectified linear unit (RELU). Some of the modules are followed by an extra max-pooling layer. The output from each module is flattened and then enters a RELU activated dense layer. Eventually, after an extra dense layer, the final output layer with 1 node can output normalized heat capacity.

**Architecture of the generator.** The architecture of the generator is shown in Fig. S5. To generate a latent vector with a desired heat capacity, the heat capacity is concatenated to a randomly generated noise vector  $z$  with a dimension of  $128 \times 1$ , which is then fed into the generator. The first five modules of the network consist of a deconvolutional layer which slides the reshaped

input with a stride of 1. The activation function is a RELU. Two modules follow the first five modules. Their number of filters is reduced from 512 to 256 and then to 128. Finally, there is a deconvolutional layer to ensure that only one structure is generated at each iteration. It means that the generator synthesizes one latent vector ( $6 \times 6 \times 2$ ) for each target value it receives. The tanh activation function is applied to the final layer for outputting continuous numbers ranging from -1 to 1.

**Architecture of the discriminator.** The architecture of the discriminator, shown in Fig. S6, has two functions. The first function is to distinguish the synthesized latent vectors from the real ones, and the second one is to determine if a generated molecule corresponds to a desired heat capacity. By appending the information of the heat capacity, a single vector rather than a high-dimension tensor is a desired format of the data as an input to the discriminator. Therefore, instead of directly feeding a real or a synthesized latent vector to the discriminator, a latent vector of a structure is first concatenated with the corresponding heat capacity. And then the concatenated vector is fed as the input to the discriminator. The discriminator is trained to distinguish the real latent vectors from the synthesized ones. The discriminator has only one intermediate dense layer with 64 nodes, followed by a RELU activation layer. The output layer is a single-node dense layer activated by the sigmoid function, which forces 0 or 1 output, indicating fake or real ones, respectively.

**Training of RRCGAN.** Before model training, we first established the training datasets. We used the interquartile range (IQR) method of identifying outliers. The reason of using IQR to establish the initial training datasets for the RRCGAN development is to exclude the outlier molecules. They are in very tails of the distribution and have no enough samples to train an accurate model. We set up a range based on the first quartile (25%),  $Q_1$ , and third quartile (75%),

$Q_3$ . To build this fence, we took 1.5 times of the IQR ( $Q_3 - Q_1$ ) and then subtracted the value from  $Q_1$  as the lower bound ( $Q_1 - 1.5 \times \text{IQR}$ ), and then added the value to  $Q_3$  ( $Q_3 + 1.5 \times \text{IQR}$ ) as the upper value. By this way, any molecule with a heat capacity value that falls outside of this range is considered an outlier. The resulted outliers only occupy 1% of the total QM9 molecules. The statistics of the 133K samples in the QM9 library is tabulated in Table S2.

The RRCGAN was designed and trained by Google's TensorFlow API. Adam was selected as the optimizer for the encoder, decoder, generator, discriminator, and regressor. The training was performed on the high-performance computing infrastructure provided by Research Computing Support Services and in part by the National Science Foundation under grant number CNS-1429294 at the University of Missouri, Columbia.

### Supplementary Note 3. Evaluation metrics

The discrepancies between the DFT calculated  $C_v$  of the generated molecules and the desired  $C_v$  and predicted  $C_v$  by the regressor, respectively, are evaluated by standard statistical metrics including the coefficient of determination ( $R^2$ ), mean absolute error (MAE), root mean squared error (RMSE) mean squared error (MSE), and relative error (RE). These metrics are provided as follows.

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \bar{y})^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (\text{S1})$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \bar{y}| \quad (\text{S2})$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2} \quad (\text{S3})$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (\text{S4})$$

$$\text{RE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \quad (\text{S5})$$

where  $y$  is the desired  $C_v$  or predicted  $C_v$  by the regressor,  $\hat{y}$  is the DFT calculated  $C_v$  of the generated molecule, and  $\bar{y}$  is the average for all the samples.  $N$  is the total number of the evaluated molecules.

#### Supplementary Note 4. Tanimoto coefficient

After converting the molecules to binary fingerprints (numerical representations), the resulted bit vectors are used to calculate the similarity coefficients. We used RDKit for all the calculations. RDKit uses a 128×128 matrices of 0 or 1 values, named bits, as fingerprints of molecules. The following formula is used to compare two molecules, A and B.

$$S_{AB} = \frac{c}{a + b - c} \quad (\text{S6})$$

where  $a$  is the bits (number of 1s) that are 1 in A,  $b$  is the bits that are 1 in B, and  $c$  is the bits that are common in both.

#### Supplementary Note 5. Distribution of properties of the generated molecules

Quality metrics are useful to assess the generated compounds. For this purpose, we compare the following quantitative properties for the generated and training molecules. The results show that the quality metrics of the generated molecules follow the distribution of the training samples.

- a. **LogP:** the octanol-water partition coefficient (log P) is a way to measure lipophilicity and has become a standard property determined for potential drugs.<sup>11</sup> For example, drugs with

high and low LogP are hydrophobic and hydrophilic, respectively. Employing Crippen estimation (atom-based calculation)<sup>11</sup> from RDKit, we compared the distribution of LogP of the generated molecules with that of the training set. As shown in Fig. S15a, LogP of the generated samples follows a similar distribution to that of the training samples.

- b. **Quantitative estimation of drug-likeness (QED):** QED is a quantitative value between 0 and 1 that represents drug-likeness based on the desirability of the druglike molecules.<sup>12</sup> Drug-likeness should be considered when selecting compounds in the early stage of drug discovery. The distribution of QED of the generated samples follows that of training samples (Fig. S15b).
- c. **Topological polar surface area (TPSA):** TPSA of a molecule is defined as the sum of surface polar atoms primarily oxygen and nitrogen atoms and their attached hydrogen atoms. TPSA is a commonly used metric in drug-like molecules that shows the ability of the drug to permeate cells.<sup>13</sup> We used RDKit to compare TPSA values for generated molecules with that of the training database. As it is shown in Fig. S15c, TPSA of generated molecules follows similar distribution compared to that of training samples.

#### **Supplementary Note 6. Dimension reduction of latent spaces.**

First, projection of the latent features is performed using principal component analysis (PCA). PCA derives components formed as a linear combination of the original variables that explain the most variance of the data. The results show that the PCA components of the latent features of the training and generated structures follow close distributions (Fig. S14a-b). We can observe well-defined boundaries of molecules in high and low values of heat capacity for mapped latent features. Second, we performed a non-linear mapping, named Spectral Embedding (SE). SE uses Laplacian

Eigenmaps to find a low dimensional representation of the latent features using a spectral decomposition of the graph Laplacian. We can test the hypothesis that the latent features can indeed explain how the model learns the structure-property relationship for catching the physical and chemical laws. Figure S14c and S14d show the spectral embedding values for the training and testing samples, respectively. For the training samples, the boundaries of the four ranges of heat capacity are clear. For the testing molecules, the very high and low values are clustered, but the middle values are mixed.

**Supplementary Tables:**

**Table S1.** Hyperparameters of RRCGAN.

Models	Hyperparameters	Values
<b>Encoder-Decoder</b>	Batch size	32
	Epochs	400
	Adam learning rate	$9 \times 10^{-5}$
	Adam regularization term ( $\beta_1$ )	0.9
<b>Regressor</b>	Batch size	128
	Epochs	120
	Adam learning rate	$1 \times 10^{-5}$
	Adam regularization term ( $\beta_1$ )	0.9
<b>Discriminator</b>	Batch size	64
	Epochs	150
	Adam learning rate	$2 \times 10^{-5}$
	Adam regularization term ( $\beta_1$ )	0.5
<b>Generator</b>	Batch size	64
	Epochs	150
	Adam learning rate	$2 \times 10^{-5}$
	Adam regularization term ( $\beta_1$ )	0.5

**Table S2.** Statistics of 133K molecules in the QM9 library.

Ave. $C_v$	Min. $C_v$	Q <sub>1</sub>	Q <sub>3</sub>	IQR	Lower bound	Upper bound	Max. $C_v$
31.6	6	28.9	34.3	5.3	20.9	42.3	47.0

Note: The unit for all the tabulated numbers is cal/(mol·K).

**Table S3.** Statistics of 63K outlier removed samples from QM9 library.

Ave. $C_v$	Min. $C_v$	Q <sub>1</sub>	Q <sub>3</sub>	IQR	Lower bound	Upper bound	Max. $C_v$
31.6	20.9	29.0	34.2	5.3	21.1	42.1	42.3

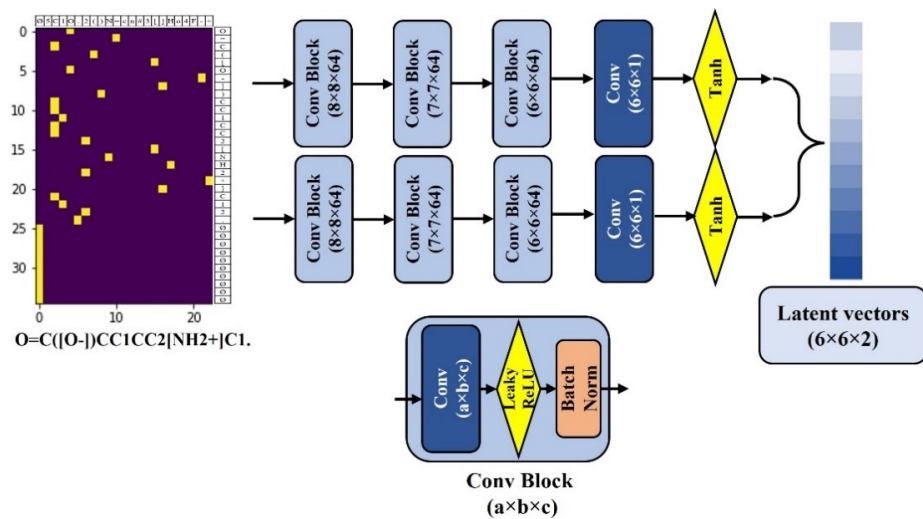
Note: The unit for all the tabulated numbers is cal/(mol·K).

**Table S4.** Hyperparameters of the XGBoost model.

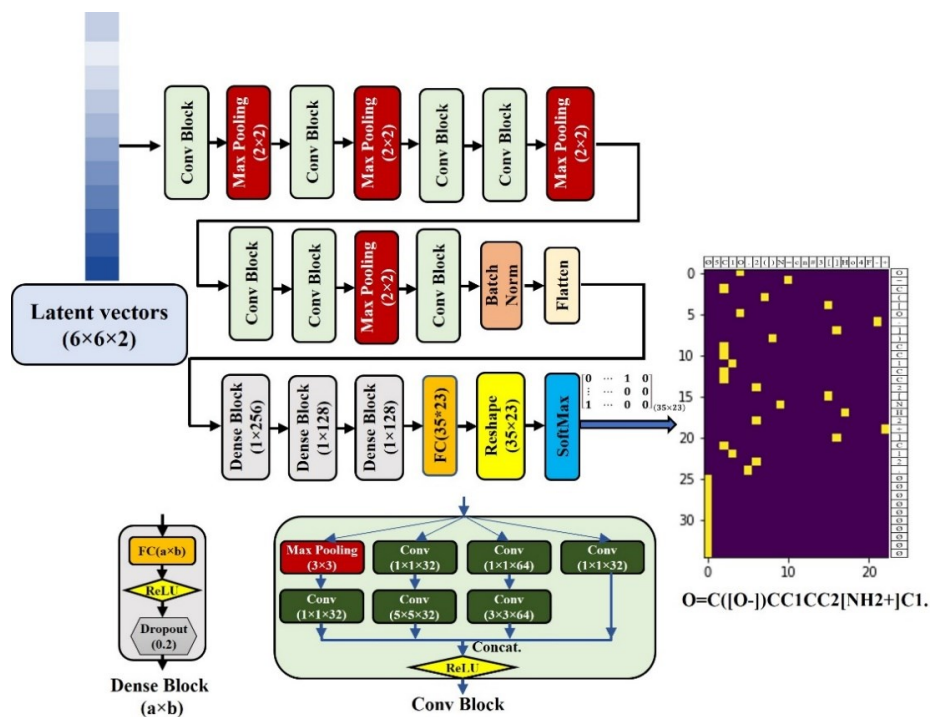
Models	Hyperparameters	Values
<b>XGBoost</b>	max_depth	4
	n_estimators	25
	min_child_weight	1
	learning_rate	0.3
	gamma	0.01
	subsample	0.7



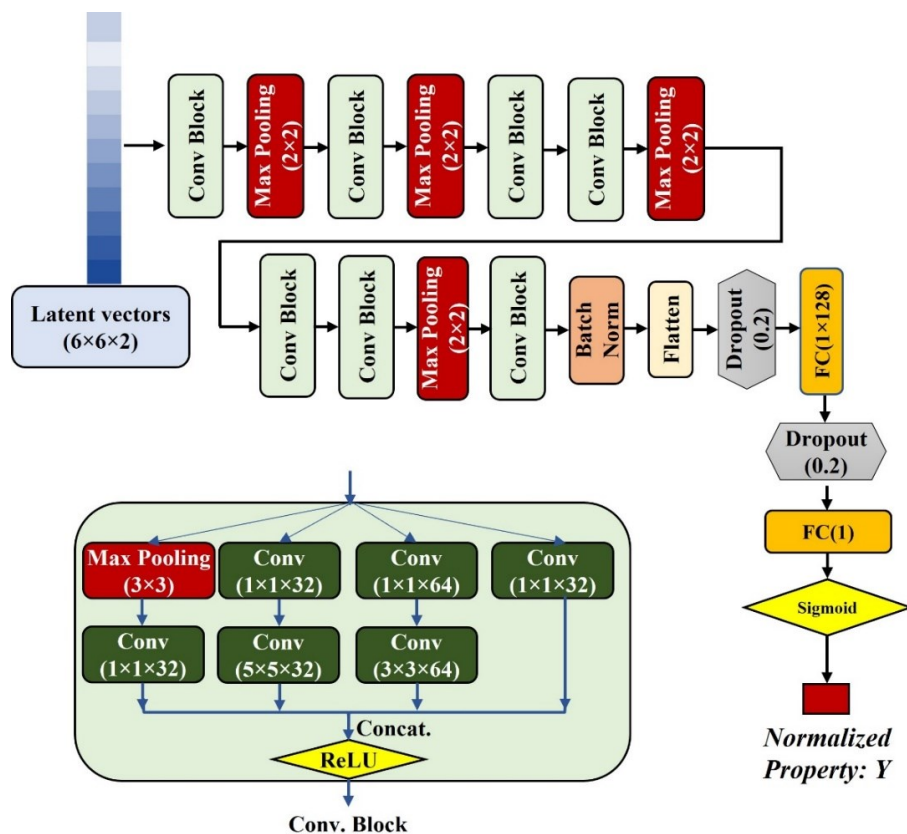
41



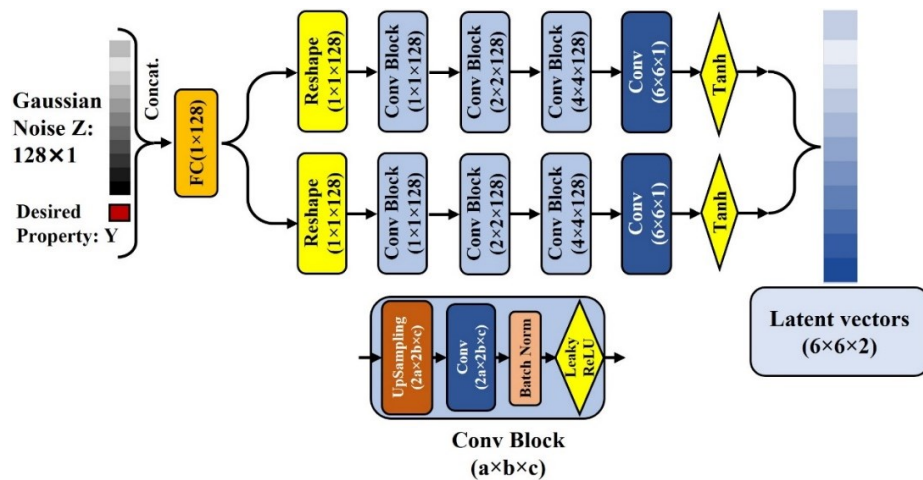
**Figure S2.** Schematic of the encoder. It takes one-hot encoded SMILES strings as input and outputs latent vectors.



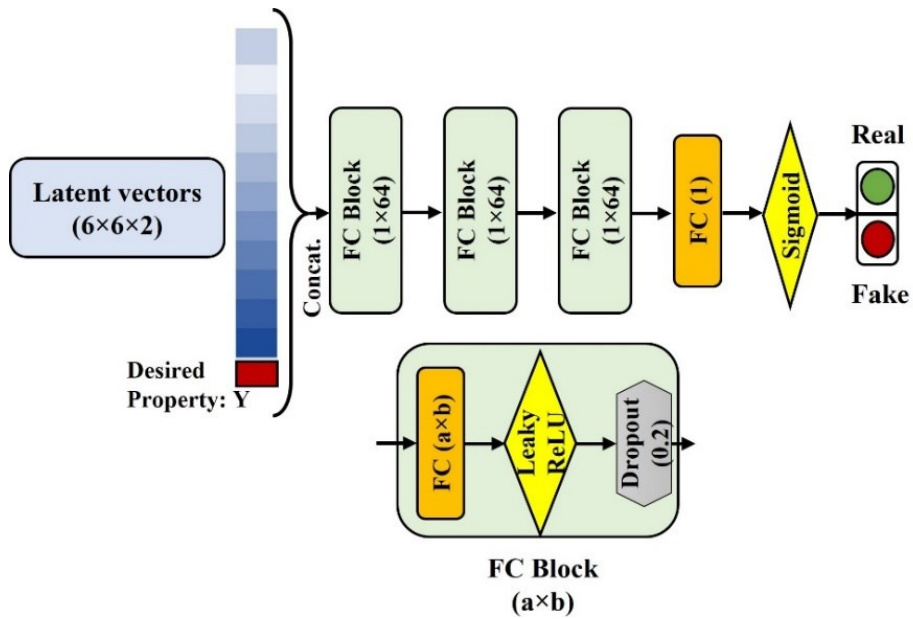
**Figure S3.** A schematic of the decoder. It takes the latent vectors as input and outputs one-hot encoded SMILES.



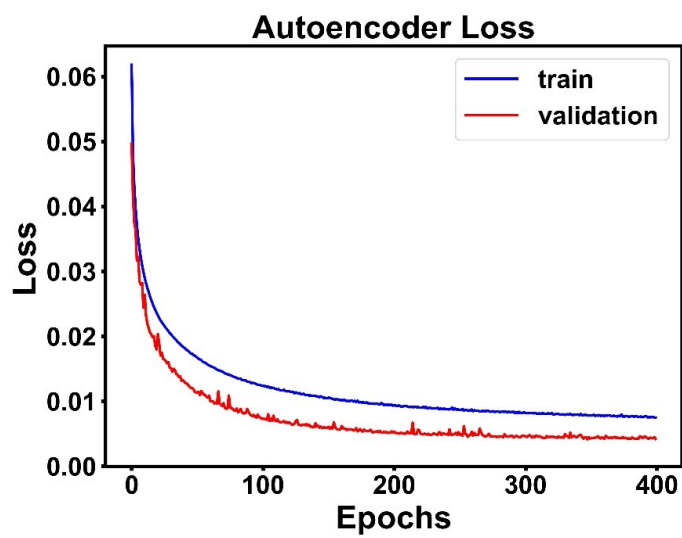
**Figure S4.** Architecture of the regressor. It takes the latent vectors as input and outputs the  $C_v$  values.



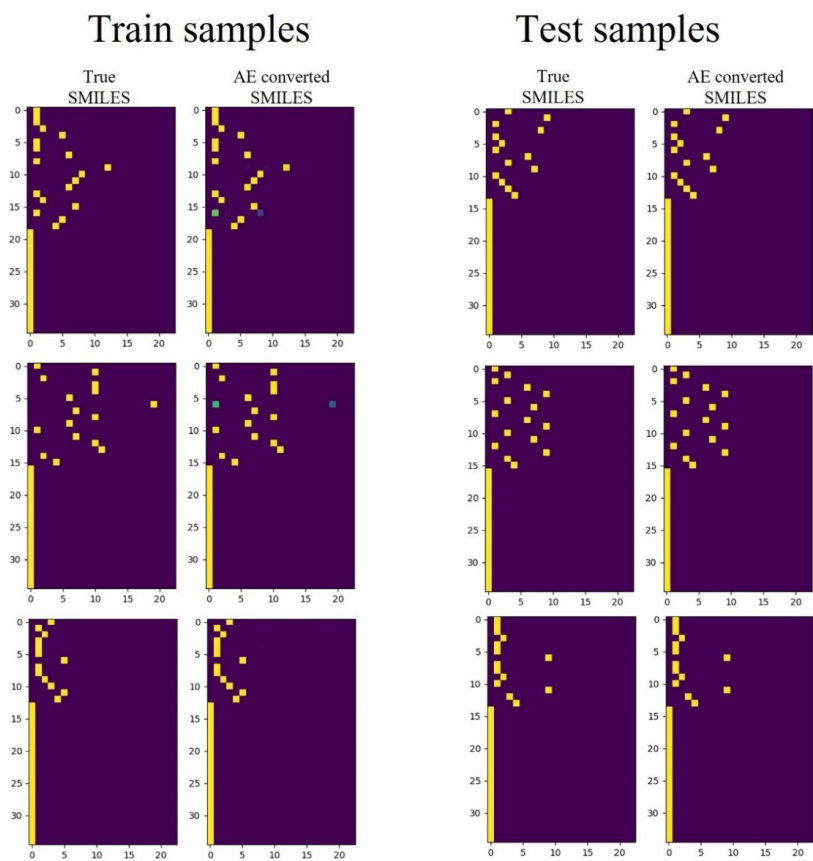
**Figure S5.** Architecture of the generator. It takes a random noise  $Z$  and a desired  $C_V$  as inputs and generates the latent features of the molecules in response to the targeted  $C_V$ .



**Figure S6.** Architecture of the discriminator. It takes a real or fake (synthesized) latent feature which is concatenated with the corresponding  $C_v$  as inputs to output either “0” for the fake or “1” for the real feature.

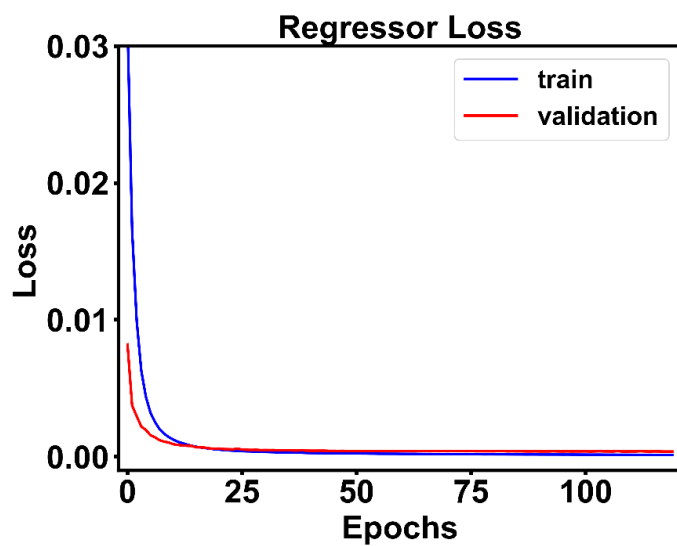


**Figure S7.** Loss evolution of the autoencoder during the training process. It shows that after 300 epochs training, the loss is stabilized, indicating a success of the training.



**Figure S8. Comparison of true one-hot encoded SMILES versus the AE encoded one-hot SMILES.** Ideally, they should be the same. Left figures show 3 training molecules, and the right figure are from 3 testing molecules. The lighter the color is, the closer probability of that string to 1 is.





**Figure S9.** Loss evolution of the regressor during the training process. It shows that after 75 epochs training, the loss is stabilized, indicating a success of the training.

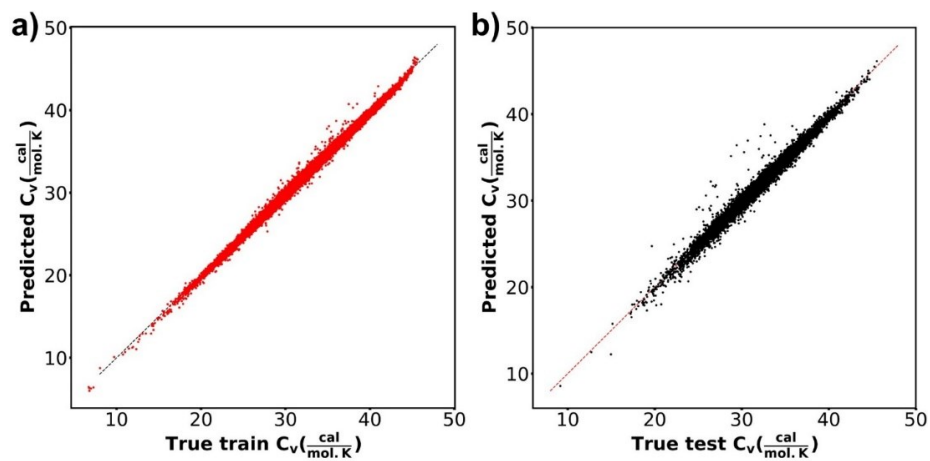
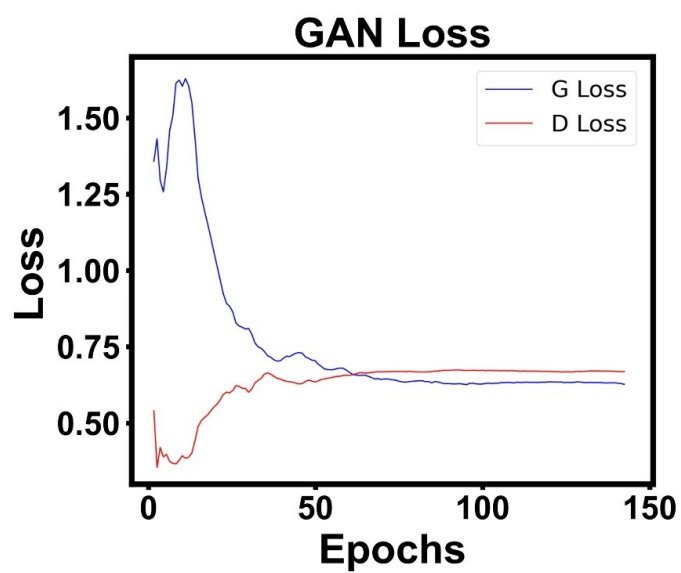
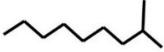
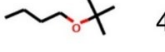

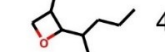
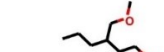
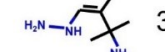
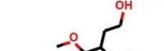


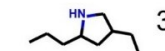


Figure S10. Distribution of the predicted  $C_v$  in  $\text{cal}/(\text{mol}\cdot\text{K})$  and the true values of the molecules in the QM9 library: (a) training molecules; (b) testing molecules.

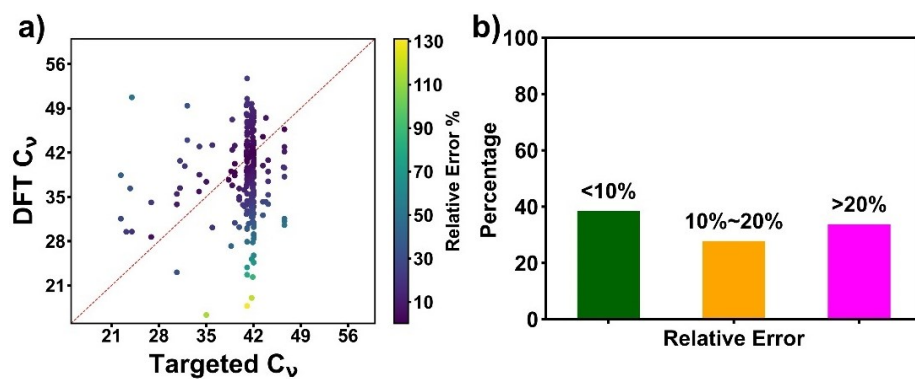


**Figure S11. Loss evolution of the generator and discriminator during the training process.**

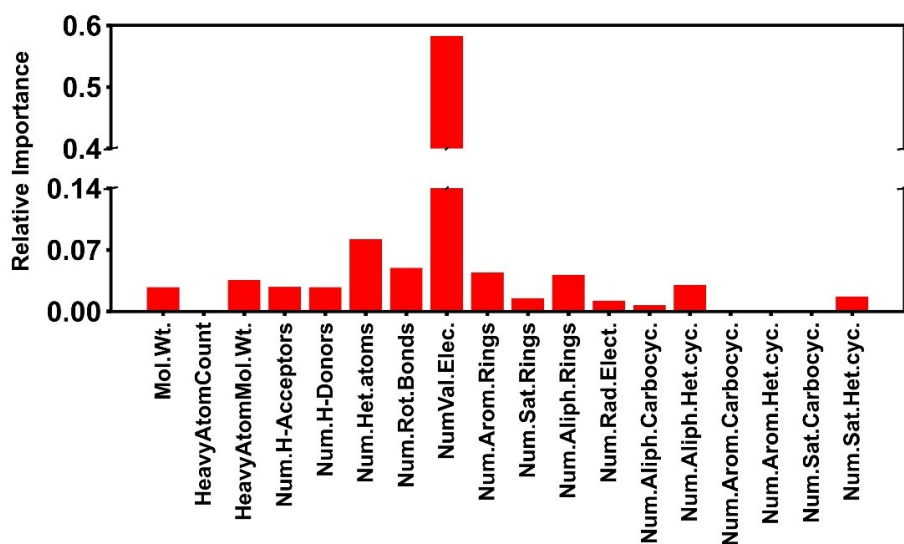
After 100 epochs, the loss is stabilized, showing a success of training.

SMILES	$C_v$ (cal/(mol·K))			SMILES	$C_v$ (cal/(mol·K))		
	tar.	pred.	DFT		tar.	pred.	DFT
	39.4	42.2	49.5		41.4	42.1	44.4
	41.6	42.1	46.6		41.8	42	44.4
	41.5	42.2	46.4		37.6	41.5	43.3
	40.0	42.1	45.2		41.8	42.2	44.1
	40.5	41.9	44.6		34.7	42.1	43.7

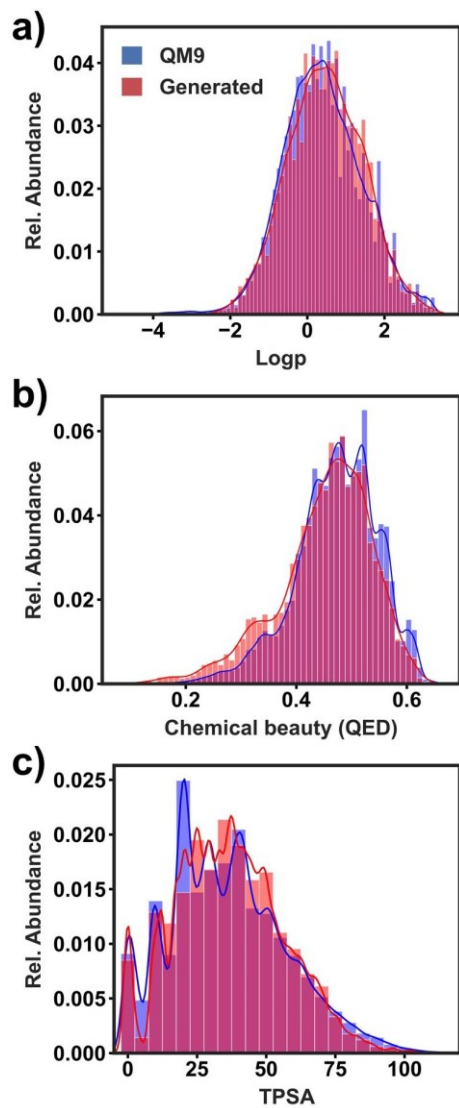
**Figure S12.** Summary of the outlier molecules with  $C_v$  of  $> 42.3$  cal/(mol·K) generated from the originally trained RRCGAN model.



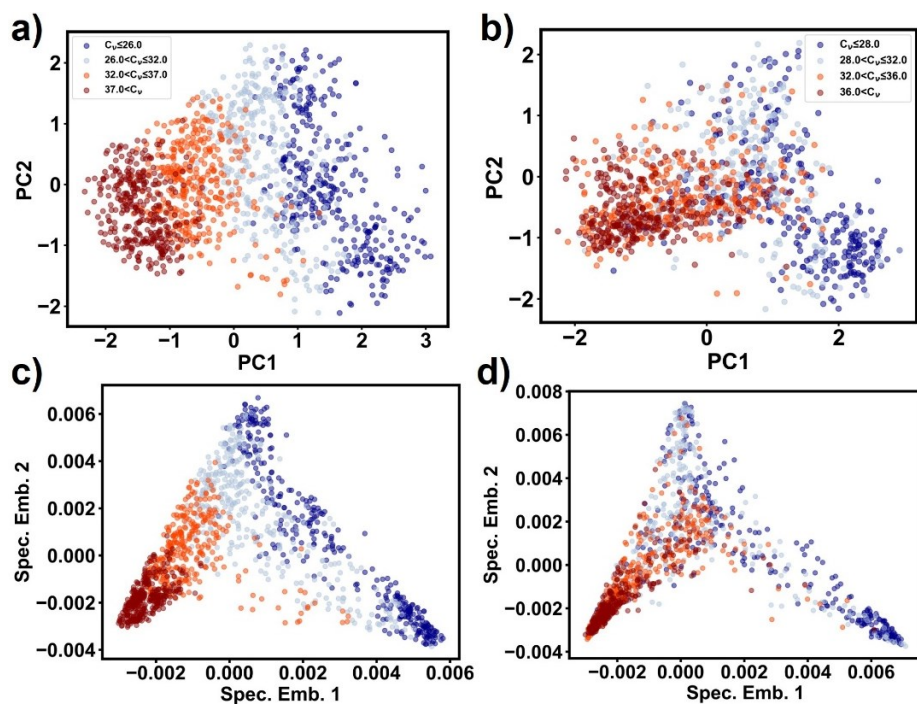
**Figure S13. Accuracy of the model trained with molecules represented by SELFIES. (a)** DFT calculated  $C_v$  of the generated molecules vs. targeted  $C_v$ . The dark blue dots have the highest accuracy. The lighters the color dots are, the less accurate the generation is. **(b)** Relative error (RE) distribution of  $C_v$  of the generated molecules.



**Figure S14.** Feature importance values of the 18 input features extracted from the well-trained random forest model. The  $R^2$  score for predicting the heat capacity using XGBoost model was 0.91 for training and 0.9 for testing. The hyperparameters of the XGBoost model are listed in Table S4. XGBoost is a prediction model that consists of assembling some weak decision trees.

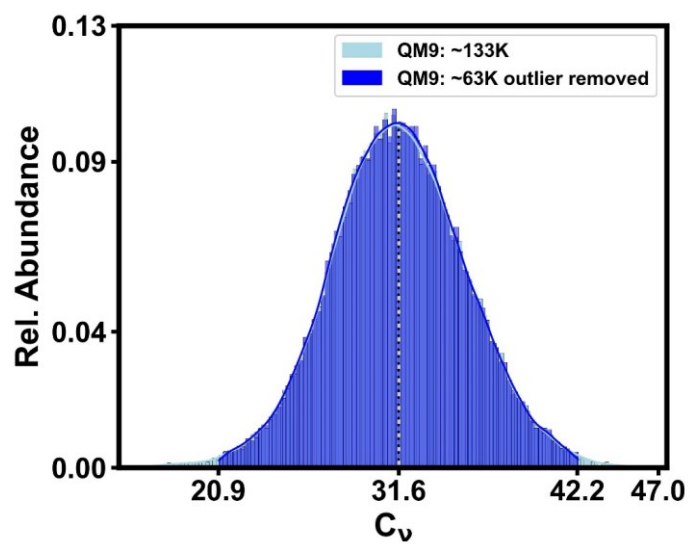


**Figure S15. Distribution of drug quality metrics for the generated samples and the training ones from the QM9 library:** (a) partition coefficient (LogP); (b) quantitative estimate of druglikeness (QED); and (c) topological polar surface area (TPSA).

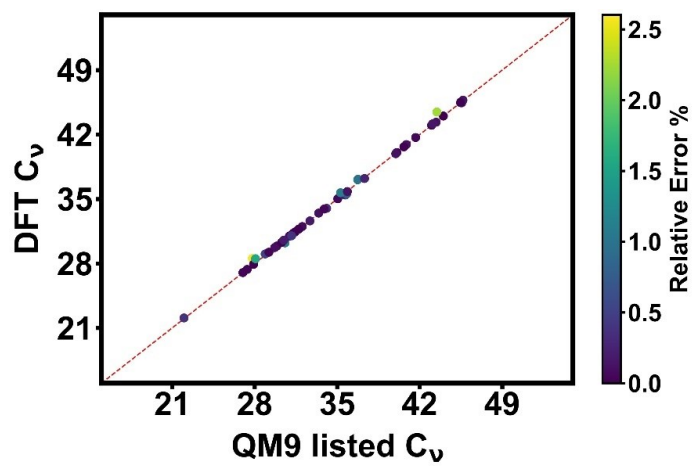


**Figure S16. Dimension reduction of the latent features output from the encoder and the generator.** PCA of the training (a) and generated molecules (b). Non-linear spectral embedding of the training (c) and generated molecules (d).





**Figure S17.** Distribution of  $C_v$  for original ~ 133K QM9 samples in light-blue color versus that of ~ 63K outlier-removed molecules sampled from original QM9. Except for the outlier regions, the two distributions are similar.



**Figure S18.** Comparison of the DFT calculated and the listed  $C_v$  values of 47 randomly selected molecules from the QM9 library. They show an MAE of 0.12 cal/(mol·K).

## Supplementary References

- 1 De Cao, N. & Kipf, T. MolGAN: An implicit generative model for small molecular graphs. Preprint at <https://arxiv.org/abs/1805.11973> (2018).
- 2 Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31-36 (1988).
- 3 Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **1**, 045024 (2020).
- 4 Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. Preprint at <https://arxiv.org/abs/1703.07076> (2017).
- 5 Duvenaud, D. K. *et al.* Convolutional Networks on Graphs for Learning Molecular Fingerprints. Preprint at <https://arxiv.org/abs/1509.09292> (2015).
- 6 Rupp, M., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **108**, 058301 (2012).
- 7 Peerless, J. S., Milliken, N. J. B., Oweida, T. J., Manning, M. D. & Yingling, Y. G. Soft Matter Informatics: Current Progress and Challenges. *Adv. Theory Simul.* **2**, 1800129 (2019).
- 8 Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **361**, 360-365 (2018).
- 9 Landrum, G. *RDKit: Open-source Cheminformatics Software*, (2006); <https://www.rdkit.org/>
- 10 O'Boyle, N. & Dalke, A. DeepSMILES: an adaptation of SMILES for use in machine-learning of chemical structures. Preprint at <https://chemrxiv.org/engage/chemrxiv/article-details/60c73ed6567dfe7e5fec388d> (2018).
- 11 Wildman, S. A. & Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Comput. Sci.* **39**, 868-873 (1999).
- 12 Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **4**, 90-98 (2012).
- 13 Pajouhesh, H. & Lenz, G. R. Medicinal chemical properties of successful central nervous system drugs. *NeuroRX* **2**, 541-553 (2005).