



OPEN Hybrid quantum-classical machine learning for generative chemistry and drug design

A. I. Gircha¹, A. S. Boev¹, K. Avchaciov², P. O. Fedichev²✉ & A. K. Fedorov¹✉

Deep generative chemistry models emerge as powerful tools to expedite drug discovery. However, the immense size and complexity of the structural space of all possible drug-like molecules pose significant obstacles, which could be overcome with hybrid architectures combining quantum computers with deep classical networks. As the first step toward this goal, we built a compact discrete variational autoencoder (DVAE) with a Restricted Boltzmann Machine (RBM) of reduced size in its latent layer. The size of the proposed model was small enough to fit on a state-of-the-art D-Wave quantum annealer and allowed training on a subset of the ChEMBL dataset of biologically active compounds. Finally, we generated 2331 novel chemical structures with medicinal chemistry and synthetic accessibility properties in the ranges typical for molecules from ChEMBL. The presented results demonstrate the feasibility of using already existing or soon-to-be-available quantum computing devices as testbeds for future drug discovery applications.

Drug design is the process of identifying biologically active compounds and relies on the efficient generation of novel, drug-like, and yet synthetically accessible compounds. So far, only about 10^8 substances have ever been synthesized¹, whereas the total number of realistic drug-like molecules is estimated to be in the range between 10^{23} and 10^{60} ². This is why deep learning³ and particularly deep generative models^{4–7} are believed to be helpful in generative chemistry and computational drug discovery applications involving sampling and scoring novel chemical structures from the very large and hitherto unknown distributions of possible drug-like molecules (see examples and benchmarks in Refs. ^{8–10}).

A fully developed generative model should implicitly estimate the fundamental molecular properties, such as stability and synthetic accessibility for each generated compound and its intermediate products. All those features depend on the ability of the network architecture to approximate the solutions of the underlying quantum mechanical problems, which is computationally hard for molecules of realistic size. Quantum computers are naturally good for solving complex quantum many-body problems¹¹ and thus may be instrumental in applications involving quantum chemistry^{12–15}. Moreover, quantum algorithms can speed up machine learning^{14,16}. Therefore, one can expect that quantum-enhanced generative models¹⁷, including quantum GANs¹⁸, may eventually be developed into ultimate generative chemistry algorithms.

Exploring the full potential of quantum machine-learning algorithms requires the development of fault-tolerant hardware¹⁶, which is not yet accessible. Meanwhile, readily available noisy intermediate-scale quantum (NISQ) devices¹⁹ provide a test-bed for the development and testing of quantum machine-learning algorithms for practical problems of modest size. For example, quantum annealing processors²⁰ could potentially enable more efficient solving quadratic unconstrained binary optimization problems and approximating sampling from the thermal distributions of transverse Ising systems. These applications are attractive in the context of machine learning as tools both for solving optimization problems^{21–24} and sampling^{25–28}. Gate-based architectures are also of interest for machine learning¹⁶, in particular, in the context of quantum GANs, which are a subject of intensive research^{29–33} including recent demonstration of learning and generation of hand-written digit images on a quantum processor³³.

In this work, we prototyped a discrete variational autoencoder (DVAE, see Ref. ³⁴), whose latent generative process is implemented in the form of a Restricted Boltzmann Machine (RBM) of a small enough size to fit readily available annealers. We trained the network on D-Wave annealer and generated 2331 novel chemical structures with medicinal chemistry and synthetic accessibility properties in the ranges typical for molecules from ChEMBL. Hence, we demonstrated that the hybrid architecture might allow practical machine-learning applications for generative chemistry and drug design. Once the hardware matures, the RBM could be turned

¹Russian Quantum Center, Skolkovo, Moscow 121205, Russia. ²Gero PTE. LTD., 133 Cecil Street #14-01 Keck Seng Tower, Singapore 069535, Singapore. ✉email: peter.fedichev@gero.ai; akf@rqc.ru

into Quantum Boltzmann Machine (QBM), and the whole system might be transformed into a Quantum VAE (QVAE³⁴) and sample from richer non-classical distributions.

Results

We proposed and characterized a generative model (see Fig. 1) in the form of a combination of a Discrete Variational Autoencoder (DVAE) model with a Restricted Boltzmann Machine (RBM) in the latent space^{34,35} and the Transformer model³⁶. The model learns good representations of chemical structures from ChEMBL, which is the manually curated database of biologically active molecules with drug-like properties³⁷.

Following Ref.⁴, we used common SMILES³⁸ encoding for organic molecules and trained the system to encode and subsequently decode molecular representations via optimizing evidence lower bound (ELBO) for DVAE log-likelihood³⁴:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\zeta}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{\zeta})] - \beta D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})). \quad (1)$$

Here, \mathbb{E} denotes the expectation value, D_{KL} is the Kullback–Leibler (KL) divergence, and $p_{\boldsymbol{\theta}}(\mathbf{z})$ is the prior distribution in the latent variable space and is encoded by RBM as in Ref.³⁴ (see MM). The two layers of RBM contain 128 units each. An RBM of this size can be sampled on readily available quantum annealers. We used the spike-and-exponential transformation³⁴ as a smoothing probability distribution between the discrete \mathbf{z} and continuous $\boldsymbol{\zeta}$ variables and employed the standard reparameterization trick to avoid calculating derivatives over random variables.

The respective encoder and decoder functions, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{\zeta})$, are approximated by the deep neural networks with Transformer layers each depending on its own set of adjustable parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. We modified the KL divergence term with $\beta = 0.1$ to avoid posterior collapse³⁹.

We trained the network for 300 epochs until apparent convergence using Gibbs sampling (see the red and yellow lines in Fig. 2 representing the total loss over the validation and train sets, respectively). In what follows, we discuss the two checkpoints: the fully trained (Gibbs-300) and, for reference purposes, the intermediate model (Gibbs-75) appearing by the end of the 75th epoch. We expect that with improvements in quantum hardware (in particular, coherence times of qubits), training the DVAE with quantum annealing technique could be comparable to or overcome existing techniques.

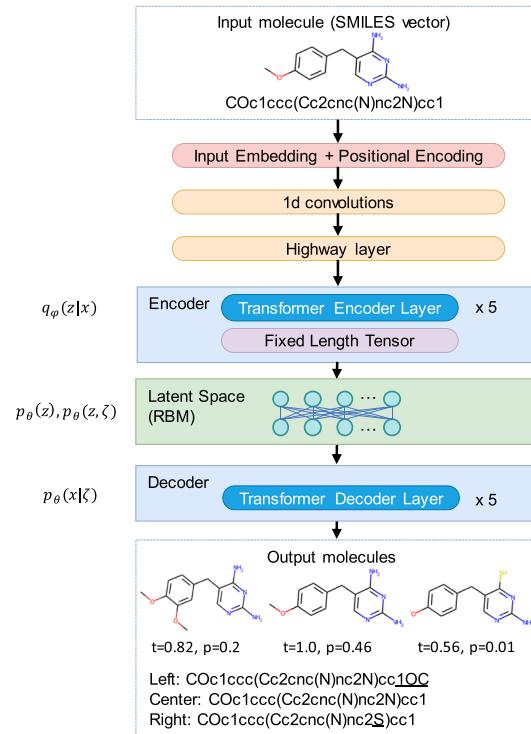


Figure 1. Scheme of the DVAE learning a joint probability distribution over the molecular structural features x and their latent variable-representations (discrete z and continuous $\boldsymbol{\zeta}$). Here, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}|\boldsymbol{\zeta})$ are the encoder and decoder distributions, respectively, whereas $p_{\boldsymbol{\theta}}(\mathbf{z})$ is the prior distribution in the latent variable space and is encoded by RBM. We provide an example of the reconstruction of a target molecule (diaveridine) using the Gibbs-300 model saved after 300 epochs of training (here $t \in [0, 1]$ is the Tanimoto similarity between the initial molecule and its reconstruction, $t = 1.0$ corresponds to perfect reconstruction, p is the output probability).

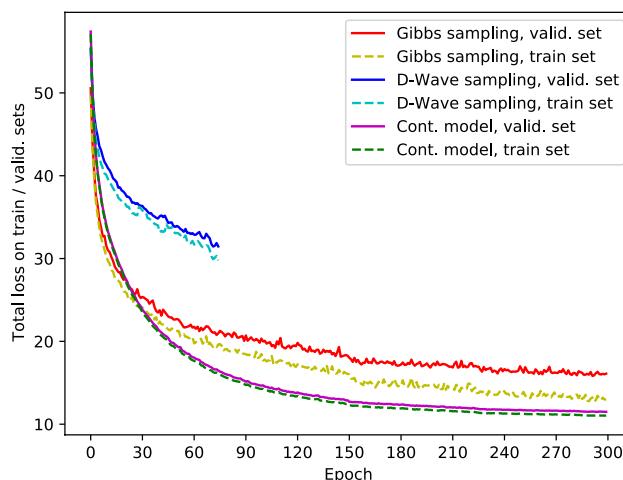


Figure 2. Learning curves of DVAE trained with classical Gibbs sampling (red, yellow) and samples from D-Wave annealer (blue, cyan). Training on D-Wave was suspended before reaching convergence due to resource limitations. Also, the learning curve of a simpler model with continuous latent variables is shown (magenta, green).

VAE is a probabilistic model. In particular, this means that each of the discrete states in the latent variables is decoded into a probability distribution of SMILES-encoded molecules. On top of Fig. 1 we provide an example of encoding a particular molecule (diaveridine) and its reconstruction by the Gibbs-300 network (see the structures at the bottom). In this case, the target molecule was reconstructed exactly in 46% runs (see the reconstruction probabilities and Tanimoto similarities to the target molecule next to the reported structures).

DVAE is a generative model that can produce novel molecules with properties that presumably match those in the training set. In Fig. 3 and Table 1, we compare the distributions of the basic biochemical properties of the molecules in the training set and among molecules generated by each of the models with discrete latent variables trained and discussed in this work. The novel molecules were mainly valid (55% and 69% in Gibbs-75 (10k molecules) and Gibbs-300 (50k molecules) models, respectively). We kept track of molecular weight (MW), the water-octanol partition coefficient (logP), the synthetic accessibility (SA⁴⁰) score, and the quantitative estimation of drug-likeness (QED⁴¹) score, which are common physico-chemical properties for benchmarking molecular generative models⁹.

Aside from the biochemical and drug-likeness properties, we also measured the novelty of generated molecules. Less than 1% of the generated molecules (0.36% and 0.22% in Gibbs-75 and Gibbs-300 models, respectively) had Tanimoto similarity larger than 0.9 to any molecule in the training set, and less than 10% of the generated molecules are similar to any molecule in the training set with $T > 0.7$ in both models. Extra training time improved both the validity of the generated molecules and brought the molecular properties closer to those found in the training set (see the relevant Gibbs-75 and Gibbs-300 columns in Table 1).

The proposed network architecture is sufficiently compact to fit the D-Wave hardware. Hence, we were able to train the network using the annealer instead of Gibbs sampling. The learning of the hybrid model on D-Wave progressed slower than that on a classical computer using Gibbs sampling (see the blue solid and cyan dashed lines in Fig. 2 corresponding to the total loss of the model on the validation and the training sets). We had, however, to stop the training before reaching convergence at the 75th epoch due to the limited performance of the available quantum hardware. With its further improvements, we expect to have the ability to prolong the

	Train set	D-Wave sampl. (75 epochs)	Gibbs sampl. (75 epochs)	Gibbs sampl. (300 epochs)
Number of samples	153,600	4290	10,000	50,000
MW	409.99/153.89	374.16/113.5	397.81/114.35	416.14/124.46
LogP	3.41/2.01	3.68/1.9	3.64/1.77	3.61/1.87
QED	0.54/0.22	0.54/0.22	0.54/0.22	0.52/0.22
SAS	2.96/0.97	3.12/0.88	3.04/0.81	3.13/0.84
Validity	1.0	0.54	0.55	0.69

Table 1. The parameters of distributions of physico-chemical properties of the molecules produced by the generative models discussed in this work. The entries in the table are mean/std values computed with the help of RDKit library⁴², where MW is molecular weight, LogP—the octanol-water partition coefficient, QED—quantitative estimation of drug-likeness, and SAS – synthetic accessibility score. Validity is a fraction of valid SMILES strings (which can be translated to a molecular graph) to a total number of generated strings.

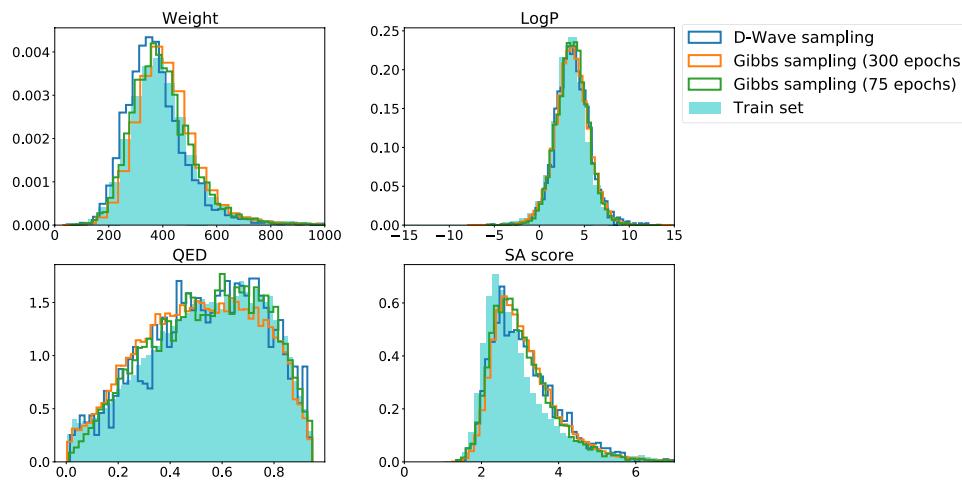


Figure 3. Distributions of physico-chemical properties of the molecules produced by the proposed generative models (same as in Table 1).

training. Eventually, we used D-Wave to generate 4290 molecular structures (2331 of which are grammatically correct, see Fig. 3 and the corresponding column in Table 1). As expected, the distributions of basic properties of the generated molecules were close to those obtained from the Gibbs-75 model and could be improved if more training time were available.

Discussion and outlook

VAEs are powerful generative machine learning models capable of learning and sampling from the unknown distribution of input data^{43,44}. As a first step towards building a hybrid quantum generative model, we prototyped the DVAE (along the lines of Ref.³⁴) with the RBM in its latent space^{34,35}. If provided with a large dataset of drug-like molecules, such a system should learn implicit rules governing the stability and synthetic accessibility of small molecules and produce useful representations of molecular structure, which could be used to generate novel and still drug-like molecules for drug design applications such as virtual screening.

As a proof of concept, we built a DVAE involving transformer layers³⁶ in the encoder and decoder components along with additional preprocessing layers that allowed our model to operate at the character-level (rather than on the word-level) to parse SMILES, the textual representations of the input molecules. Using SMILES is not necessarily the best option since these strings are not 100% valid. The only property of SMILES that is essential in our approach is that it is a representation of molecules in terms of character strings and hence we believe that DVAEs can be built to operate with alternative character string representations of molecules, such as SELFIES⁴⁵.

We trained a compact DVAE with the RBM consisting of two layers of just 128 units each on a small subset containing almost 200,000-random molecules from the ChEMBL database of manually curated and biologically active molecules as the training set. On classical hardware, the system could be trained with Gibbs sampling. We were able to show that the training converged and used the network to generate molecules with the distribution of the basic properties, such as logP, and QED, closely matching those in the training set. Simultaneously, the average size of the molecules increased as the training of the network was progressing. There are relatively harder-to-synthesize compounds among the molecules generated by the network⁴.

Our generative model outputs drug-like molecules and may be deployed on already existing quantum annealing devices (such as D-Wave Advantage processor). Training of the same architecture network on the quantum annealer proceeded slower per epoch than on the classical computer, most probably due to noise. Nevertheless, the distributions of the molecular properties of generated molecules were sufficiently close to those in the training set or among the molecules generated by classical counterparts Gibbs-75 and 300. While certain discrepancies between distributions were present, these results have been computed only after a limited number of training epochs due to the restrictions on public access to the quantum computer.

Computational drug design applications depend on but are not limited to the generation of novel and synthetically accessible molecules, which is the focus of this work. The authors of the original paper⁴ have already proposed training additional properties, such as the prediction of the binding constant to a particular target on top of the autoencoder loss. Although a direct extension of VAE for these tasks may be challenging and require further refinements⁶, in such a form, the network could be used in problems involving actual drug design, i.e., for generating of novel compounds binding specific medically relevant targets. We did not attempt to demonstrate such a capability. However, we have no doubts that DVAE and, eventually, its hybrid implementations, such as QVAE, can be appropriately refitted by adding the extra loss.

The RBM could be turned into Quantum Boltzmann Machine (QBM) so that the whole system might be transformed into a Quantum VAE (QVAE,³⁴) and sample from potentially richer non-classical distributions. Using genuine QBMs should speed up the training of the system ($\mathcal{O}(\log N)$ vs. $\mathcal{O}(\sqrt{N})$ with N being the size of the network¹⁶). There was a demonstration in Ref.³⁴, where “quantum” samplers with the non-vanishing transverse fields outperformed DVAE if assessed by metrics achieved at the same number of training cycles (epochs).

Construction of QVAE with the controllable non-zero transverse field can, in principle, be performed on the existing generation of D-Wave chips. However, it would require additional hardware tuning and applying a combination of extra tricks such as reverse anneal schedule, pause-and-quench, etc⁴⁶.

We demonstrated that a useful VAE can be built and trained to generate drug-like molecules while keeping the size of latent representation small and hence practically attainable on already existing quantum annealing devices. We expect that with further developments in the engineering of quantum computing devices, hybrid architectures similar to QVAE would surpass their classical counterparts. More specifically, the network architecture proposed in this work may provide the baseline for further refinements required for running genuinely quantum generative models. The benefit may be especially large in problems potentially involving rules of quantum chemistry, such as learning efficient representations of molecular structures for applications related to generative chemistry and drug design.

Methods

We proposed and characterized classical and quantum annealer models, which are a combination of Discrete Variational Autoencoder (DVAE) with Restricted Boltzmann Machine (RBM) in the latent space^{34,35} and the Transformer model³⁶. Original Transformer model was proposed for word-level natural language processing tasks and has encoder-decoder architecture. We used original Transformer layers and developed additional preprocessing layers that allowed us to process character-level SMILES descriptions of molecules. We trained the proposed models on a subset of the ChEMBL dataset by optimizing evidence lower bound (ELBO) for DVAE log-likelihood³⁴, modified with additional coefficient β that multiplies KL divergence term³⁹, see Eq. (1). The sketch of the architecture of our models is illustrated in Fig. 1.

Below we describe in details the dataset, the network architecture, the training parameters, and the training schedule of the classical and quantum annealer models. Also, we describe a simpler classical model with continuous latent variables, which we used in the experiment shown in Fig. 2.

Dataset. We used a subset of molecules from the ChEMBL (release 26) database^{47,48}. Our dataset consisted of the 192,000 structures encoded by SMILES strings of the maximum length of 200 symbols and containing the atoms from the organic subset only (B, C, N, O, P, S, F, Cl, Br, I). To focus on the relevant biologically active compounds, we removed salt residuals. Finally, we converted all SMILES into the canonical format with the help of RDKit⁴².

The processed molecules were randomly assigned into train and validation sets each containing 80% and 20% of all samples (153,600 and 38,400 molecules), respectively.

Training DVAE using Gibbs-sampling. Molecular SMILES strings are tokenized with the regular expression from Ref.⁴⁹, which produced 42 unique tokens. Standard trainable embedding layer and positional encoding from Ref.³⁶ are used. Our implementation utilized a combination of embedding and positional encoding, in which positional encoding is multiplied by an additional correction factor:

$$\tilde{\mathbf{x}}_{emb} = \sqrt{d_{emb}} \mathbf{x}_{emb} + \frac{1}{\sqrt{d_{emb}}} \mathbf{pe}, \quad (2)$$

where \mathbf{x}_{emb} is embedding tensor, \mathbf{pe} is positional encoding tensor and d_{emb} is the dimensionality of the embedding. This factor is required to make the proportion between embedding tensor and positional encoding closer to that in the original model³⁶. The dimension of embeddings is a model hyperparameter which was set to 32.

We employed a layer of one-dimensional convolutions and a highway layer⁵⁰ as additional preprocessing layers between the embedding layer and the encoder component. The convolution layer with 160 filters and the kernel size equal to 5 was developed based on Ref.⁵¹. We used highway layers since such layers have been shown to improve the quality of character-level models^{51,52}.

The preprocessed 160-dimensional tensor is passed from the highway layer to the encoder, consisting of the stack of 5 Transformer encoder layers. The width of the feed-forward part of the layers is equal to 320. The number of heads in Multi-Head attention is 10. We used GeLU activation⁵³ functions and Dropout with the rate of 0.1.

Original Transformer encoder layers produce output tensor of variable length. The length of the tensor is equal to the size of the input string. In order to further reduce the dimensionality of the latent space layer in the model, we construct a fixed-length tensor from the Transformer encoder output tensor \mathbf{u} by calculating the fixed number of vectors from \mathbf{u} , which we then concatenate in one tensor. The first two of these vectors are the vector with index 0 from Transformer layers output \mathbf{u} and the vector equal to the arithmetic mean of all vectors along the length of the tensor \mathbf{u} . Next, we consider the subsets S_n^m , each consisting of vectors with indices that have the same remainder after division by n for $n = 2, 3, 4, 5$:

$$S_n^m = \{\mathbf{u}_i : i \equiv m \pmod{n}\}, m = 0, \dots, n-1.$$

For each S_n^m , we compute the arithmetic mean and concatenate all calculated vectors into the fixed-length output tensor.

Restricted Boltzmann machine (RBM) is implemented in the latent space as presented in papers^{34,35}. The probability distribution of RBM is

$$p_{\theta}(\mathbf{z}) \equiv e^{-E_{\theta}(\mathbf{z})} / Z_{\theta}, Z_{\theta} \equiv \sum_{\mathbf{z}} e^{-E_{\theta}(\mathbf{z})}, E_{\theta}(\mathbf{z}) = \sum_l z_l h_l + \sum_{l < m} W_{lm} z_l z_m, \mathbf{h}, \mathbf{W} \in \{\theta\},$$

where h_l are bias weights for units z_l and each W_{lm} is the weight associated with the connection between units z_l and z_m . The effective temperature is supposed to be equal to 1.0 and is not presented in the formulas. RBM in the proposed model consists of two layers of 128 units each. RBM of this size can be sampled using existing quantum annealing devices. It is worth noting that all the units of RBM in DVAE are latent variables and connected to the rest of the model. Hence, there is no distinction between “hidden” and “visible” units as for standalone RBM^{34,35}.

An informal description of the internal working of the model in the latent space is as follows. The output of the encoder is the vector of probabilities of the discrete latent variables z_i being equal to 1, which are conditioned on the input x of the model. These probabilities are sampled to obtain latent binary vector z . Continuous variables ζ are sampled using spike-and-exponential smoothing probability distribution $r(\zeta|z)$ ³⁴. Vector ζ is used as an input to the decoder module. During training, the parameters of the RBM are adjusted in order to memorize the statistics of the binary vectors z that appear in the latent space. The calculation of the gradient of the parameters of the RBM consists of two parts: the so-called “positive” and “negative” phases. The “positive” phase is calculated using the backpropagation algorithm after the application of the reparameterization trick, which is used to avoid calculating of derivatives over random variables. The “negative” phase of the gradient is estimated using sampling from the RBM distribution.

For molecule reconstruction or generation of similar molecules to a given molecule, the preprocessed SMILES description of the given molecule is passed to the input of the encoder and the whole model is executed. An example of molecule reconstruction and generation of similar molecules is depicted in Fig. 1. For generation of an entirely new molecule the encoder is not used, the trained RBM is sampled to obtain latent binary vector z . This vector is then used to calculate the latent vector of continuous variables ζ , which is given as an input to the decoder. Table 1 and Fig. 3 show results for newly generated molecules.

RBM is sampled by performing 30 steps of Gibbs updates using persistent contrastive divergence (PCD)⁵⁴.

The decoder works in two modes: training and inference (generation). In the inference mode, decoder uses preprocessing layers. The main part of data processing in both training and inference modes of the decoder consists of Transformer decoder layers. Altogether, we used 5 Transformer decoder layers of the size $d_{model} = 160$ (GeLU activation, dropout = 0.1). The width of the feed-forward part of the layers was equal to 320, and the number of heads in Multi-Head attention was 10.

To train the model, we used the rebalanced objective function, in which the KL divergence term is multiplied by the additional coefficient $\beta = 0.1$ ³⁹ to avoid the posterior collapse problem, and employed the Adam optimizer.

In contrast to the original Transformer model, we used a different learning rate schedule: we trained the model for 300 epochs using the MultiStep learning rate schedule with the initial learning rate equal to 6×10^{-5} . The learning rate was subsequently reduced by the factor of 0.5 at points corresponding to 50%, 75%, and 95% of the length of the training process.

For estimation of the logarithm of the partition function of Boltzmann distribution, we used annealed importance sampling (AIS) algorithm⁵⁵ during the evaluation of the model at the end of each epoch using 10 intermediate distributions and 500 samples.

Due to resource constraints, we did not have a chance to optimize the hyperparameters or too many architectural variants of the model. The presented variant of the network just worked and can be considered the first step toward a real and effective solution.

Training DVAE on a quantum annealer. We used exactly the same network architecture on the quantum annealer with the only difference from the classical case being that the RBM in the latent space was sampled using D-Wave Advantage processor. Also, the quantum model was trained during 75 epochs with constant learning rate equal to 6×10^{-5} .

For estimating the logarithm of the partition function of the Boltzmann distribution during the evaluation of the model, we used a different version of annealed importance sampling (AIS; see Ref.⁵⁶) with the same parameters as in the classical case.

Training model with continuous latent variables. The model with continuous variables in the latent space has similar architecture to the discrete one but is smaller in size. The latent space contains 32 + 32 normally distributed continuous random variables.

The preprocessing convolution layer consists of 100 filters with kernel size equal to 5. The encoder/decoder consists of 2 Transformer encoder/decoder layers with the width of feed-forward part equal to 200.

The fixed length tensor is calculated in the similar way as in the discrete model. The model is trained using the same initial learning rate and learning rate schedule as in the discrete case.

Calculation of molecular similarity with fingerprint. Fingerprints for each molecule are generated using a default function `RDKitFingerprint` in RDKit⁴². This algorithm produces a topological fingerprint represented by a bit vector with the size of 2048 bits. The Tanimoto similarity is known as a reasonable metric for matching molecules sharing similar fragments⁵⁷ and is defined for two fingerprints a, b as:

$$T(a, b) = \frac{C}{A + B - C} \quad (3)$$

where C is a number of common non-zero bits in a and b ; A and B are numbers of non-zero bits that are present in a and b , respectively. The Tanimoto distance could be defined as $D(a, b) = 1 - T(a, b)$. From the definition, it follows that completely similar molecules (shared identical set of fragments) have Tanimoto similarity equal to 1, while dissimilar molecules (no common fragments) have $T = 0$.

Code availability

The code that is deemed central to the conclusions is available from the corresponding author upon reasonable request. The data that supports the findings of this study (generated molecules) are available at <https://doi.org/10.5281/zenodo.7827952>

Received: 25 October 2022; Accepted: 31 March 2023

Published online: 22 May 2023

References

1. Kim, S. *et al.* Pubchem substance and compound databases. *Nucl. Acids Res.* **44**, D1202–D1213 (2016).
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. Deng, J., Yang, Z., Ojima, I., Samaras, D. & Wang, F. Artificial intelligence in drug discovery: Applications and techniques. *Brief. Bioinform.* **23**, bbab430 (2022).
4. Gómez-Bombarelli, R. *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
5. 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. arXiv preprint [arXiv:1705.10843](https://arxiv.org/abs/1705.10843) (2017).
6. Griffiths, R.-R. & Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.* **11**, 577–586 (2020).
7. Zhavoronkov, A. *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **37**, 1038–1040 (2019).
8. Brown, N., Fiscato, M., Segler, M. H. & Vaucher, A. C. Guacamol: Benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **59**, 1096–1108 (2019).
9. Polykovskiy, D. *et al.* Molecular sets (moses): A benchmarking platform for molecular generation models. *Front. Pharmacol.* **11**, 1931 (2020).
10. Molecular graph generation on zinc. <https://paperswithcode.com/sota/molecular-graph-generation-on-zinc>.
11. Lloyd, S. Universal quantum simulators. *Science* **273**, 1073–1078 (1996).
12. McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C. & Yuan, X. Quantum computational chemistry. *Rev. Mod. Phys.* **92**, 015003. <https://doi.org/10.1103/RevModPhys.92.015003> (2020).
13. Bauer, B., Bravyi, S., Motta, M. & Chan, G.K.-L. Quantum algorithms for quantum chemistry and quantum materials science. *Chem. Rev.* **120**, 12685–12717. <https://doi.org/10.1021/acs.chemrev.9b00829> (2020).
14. Cao, Y., Romero, J. & Aspuru-Guzik, A. Potential of quantum computing for drug discovery. *IBM J. Res. Dev.* **62**, 6–1. <https://doi.org/10.1147/JRD.2018.2888987> (2018).
15. Fedorov, A. K. & Gelfand, M. S. Towards practical applications in quantum computational biology. *Nat. Comput. Sci.* **1**, 114–119. <https://doi.org/10.1038/s43588-021-00024-z> (2021).
16. Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202. <https://doi.org/10.1038/nature23474> (2017).
17. Gao, X., Anschuetz, E. R., Wang, S.-T., Cirac, J. I. & Lukin, M. D. Enhancing generative models via quantum correlations. *Phys. Rev. X* **2101**, 08354 (2021).
18. Li, J., Topaloglu, R. & Ghosh, S. Quantum generative models for small molecule drug discovery. arXiv preprint [arXiv:2101.03438](https://arxiv.org/abs/2101.03438) (2021).
19. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2**, 79. <https://doi.org/10.22331/q-2018-08-06-79> (2018).
20. Johnson, M. W. *et al.* Quantum annealing with manufactured spins. *Nature* **473**, 194–198. <https://doi.org/10.1038/nature10012> (2011).
21. Neven, H., Denchev, V. S., Rose, G. & Macready, W. G. *Training a binary classifier with the quantum adiabatic algorithm* arXiv preprint [arXiv:0811.0416](https://arxiv.org/abs/0811.0416) (2008).
22. Denchev, V. S., Ding, N., Vishwanathan, S. V. N. & Neven, H. *Robust classification with adiabatic quantum optimization* arXiv preprint [arXiv:1205.1148](https://arxiv.org/abs/1205.1148) (2012).
23. Pudenz, K. L. & Lidar, D. A. Quantum adiabatic machine learning. *Quant. Inf. Process.* **12**, 2027–2070. <https://doi.org/10.1007/s11128-012-0506-4> (2013).
24. Mott, A., Job, J., Vlimant, J.-R., Lidar, D. & Spiropulu, M. Solving a Higgs optimization problem with quantum annealing for machine learning. *Nature* **550**, 375–379. <https://doi.org/10.1038/nature24047> (2017).
25. Raymond, J., Yarkoni, S. & Andriyash, E. Global warming: Temperature estimation in annealers. *Front. ICT* **3**, 23. <https://doi.org/10.3389/fict.2016.00023> (2016).
26. Perdomo-Ortiz, A., Benedetti, M., Realpe-Gómez, J. & Biswas, R. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quant. Sci. Technol.* **3**, 030502. <https://doi.org/10.1088/2058-9565/aab859> (2018).
27. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B. & Melko, R. Quantum Boltzmann machine. *Phys. Rev. X* **8**, 021050. <https://doi.org/10.1103/PhysRevX.8.021050> (2018).
28. Dixit, V., Selvarajan, R., Alam, M. A., Humble, T. S. & Kais, S. Training restricted Boltzmann machines with a d-wave quantum annealer. *Front. Phys.* **9**, 374. <https://doi.org/10.3389/fphy.2021.589626> (2021).
29. Lloyd, S. & Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **121**, 040502. <https://doi.org/10.1103/PhysRevLett.121.040502> (2018).
30. Dallaire-Demers, P.-L. & Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **98**, 012324. <https://doi.org/10.1103/PhysRevA.98.012324> (2018).
31. Romero, J. & Aspuru-Guzik, A. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *Adv. Quant. Technol.* **1901**, 00848 (2019).
32. Zoufal, C., Lucchi, A. & Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quant. Inf.* **5**, 103. <https://doi.org/10.1038/s41534-019-0223-2> (2019).
33. Huang, H.-L. *et al.* Experimental quantum generative adversarial networks for image generation. *Phys. Rev. Appl.* **16**, 024051. <https://doi.org/10.1103/PhysRevApplied.16.024051> (2021).
34. Khoshaman, A. *et al.* Quantum variational autoencoder. *Quant. Sci. Technol.* **4**, 014001 (2018).
35. Rolfe, J. T. Discrete variational autoencoders. In *5th International Conference on Learning Representations, ICLR 2017-Conference Track Proceedings* (2016). [http://arxiv.org/abs/1609.02200](https://arxiv.org/abs/1609.02200).
36. Vaswani, A. *et al.* Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010 (Curran Associates Inc., 2017).
37. Gaulton, A. *et al.* The ChEMBL database in 2017. *Nucl. Acids Res.* **45**, D945–D954 (2017).
38. 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).

39. Yan, C., Wang, S., Yang, J., Xu, T. & Huang, J. Re-balancing variational autoencoder loss for molecule sequence generation. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '20 (Association for Computing Machinery, 2020). <https://doi.org/10.1145/3388440.3412458>.
40. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1**, 1–11. <https://doi.org/10.1186/1758-2946-1-8> (2009).
41. Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **4**(2), 90–98 (2012).
42. Rdkit: Open-source cheminformatics software. <https://www.rdkit.org/>.
43. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013).
44. Kingma, D. P. & Welling, M. An introduction to variational autoencoders. arXiv preprint [arXiv:1906.02691](https://arxiv.org/abs/1906.02691) (2019).
45. 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).
46. D-wave system documentation. https://docs.dwavesys.com/docs/latest/c_qpu_0.html.
47. Gaulton, A. *et al.* ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **40**, D1100–D1107. <https://doi.org/10.1093/nar/gkr777> (2011).
48. Gaulton, A. *et al.* The ChEMBL database in 2017. *Nucleic Acids Res.* **45**, D945–D954 (2017).
49. Schwaller, P., Gaudin, T., Lányi, D., Bekas, C. & Laino, T. “found in translation”: Predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem. Sci.* **9**, 6091–6098. <https://doi.org/10.1039/C8SC02339E> (2018).
50. Srivastava, R. K., Greff, K. & Schmidhuber, J. Highway Networks arXiv preprint [arXiv:1505.00387v2](https://arxiv.org/abs/1505.00387v2) (2015).
51. Banar, N., Daelemans, W. & Kestemont, M. Character-level transformer-based neural machine translation. In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, NLPiR 2020, 149–156 (Association for Computing Machinery, 2020). <https://doi.org/10.1145/3443279.3443310>.
52. Kim, Y., Jernite, Y., Sontag, D. & Rush, A. M. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 2741–2749 (AAAI Press, 2016).
53. Hendrycks, D. & Gimpel, K. Gaussian Error Linear Units (GELUs) [arXiv:1606.08415v4](https://arxiv.org/abs/1606.08415v4) (2016).
54. Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, 1064–1071 (Association for Computing Machinery, 2008). <https://doi.org/10.1145/1390156.1390290>.
55. Neal, R. Annealed importance sampling. *Stat. Comput.* **11**, 125–139 (2001).
56. Ulanov, A. E., Tiunov, E. S. & Lvovsky, A. I. *Quantum-inspired annealers as boltzmann generators for machine learning and statistical physics* [arXiv:1912.08480](https://arxiv.org/abs/1912.08480) (2019).
57. Bajusz, D., Rácz, A. & Héberger, K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?. *J. Cheminform.* **7**, 1–13. <https://doi.org/10.1186/s13321-015-0069-3> (2015).

Acknowledgements

We thank S. Usmanov for help in performing the experiment. P.O.F and A. K. would like to thank Dr. A. Tarkhov from Gero and Daria Orhunova for fruitful discussion and help with the data analysis. We acknowledge online cloud access to the quantum annealing device produced by D-Wave Systems (all experiments with D-wave hardware were performed within the program of global response to COVID-19 and completed in 2021). A.K.F. thanks Russian Science Foundation grant (19-71-10092). P.O.F. and K. A. are supported by Gero PTE. LTD. (Singapore).

Author contributions

All the authors jointly developed the problem statement and analyzed existing state-of-the-art techniques. A.I.G. and A.S.B. implemented the considered methods and performed the simulation of molecules. All the authors contributed to discussions of the results and writing the manuscript. P.O.F. and A.K.F. supervised the project.

Competing interests

Owing to the employments and consulting activities of A.I.G., A.S.B., and A.K.F., they have financial interests in the commercial applications of quantum computing. P.O.F. and K.A. are employees of (P.O.F. is a stake holder in) Gero PTE. LTD. and hence may have financial interests in commercial applications of quantum or hybrid AI/ML systems for drug discovery.

Additional information

Correspondence and requests for materials should be addressed to P.O.F. or A.K.F.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023