MatVAE: Independently Trained Nested Variational Autoencoder for Generating Chemical Structural Formula

Yoshihiro Osakabe,[1] **Akinori Asahara**[1]
[1] Hitachi, Ltd. Research and Development Group
yoshihiro.osakabe.fj@hitachi.com, akinori.asahara.bq@hitachi.com

Abstract

Rapid materials development utilizes deep generative models to suggest candidate compounds with desirable properties before actual experiments. Such models successfully generate novel candidates with improved properties in some cases, but they usually require a large experimental dataset which is difficult to obtain. We propose MatVAE–two nested VAEs independently trained on different datasets. The first VAE, which is trained on a huge open dataset, is a universal generator of chemical structural formulae, and the second VAE, which is trained on a small experimental dataset, learns the structure–property relation. This training framework can be understood as a semi-supervised learning, which is expected to enhance model transferability. We verified that MatVAE generates five times more valid candidate compounds than the conventional un-nested single VAE model.

Introduction

Determining the optimal combination of ingredients and parameters can be a costly and time-consuming process in product development. Materials informatics (MI) is an emerging field that integrates informatics and materials science with the goal of greatly reducing the resources and risks involved in discovering, investing in, and deploying new materials (Curtarolo et al. 2013). Recently, artificial intelligence (AI) has led to improvements in MI; experimental candidates can be narrowed down without unnecessary trial and error before actual experiments to discover or create new materials with desirable property values.

Statistically modeling the relationship between descriptors and property values of a compound is a widely used method for predicting the property values of candidate compounds without conducting experiments. In drug discovery, this relationship is referred to as the quantitative structure–activity relationship (QSAR). In general, such methodology is based on two processes; calculating descriptors from the chemical structure by extracting structural features and building a model integrating the descriptors and the property values. The constructed statistical model can be used to predict property values from the chemical structures. Recently, machine learning techniques have been widely used to build statistical models. However, there are two problems when solving the inverse problem to obtain the chemical structure with desired properties. The first problem is that a statistical model is generally a complex nonlinear function, so an inverse function that yields a descriptor from a property value cannot be obtained explicitly. The second problem is that even if a descriptor is obtained, it is difficult to construct a chemical structure containing that descriptor.

In recent years, an approach that has been extensively studied is to use deep generative models to directly obtain chemical structures that have desirable properties without calculating the descriptors. Previously, methods for training a generative adversarial network (GAN) with the reinforcement learning (RL) framework have been reported (Olivecrona et al. 2017; De Cao and Kipf 2018). The major difference between them is the representation of the compounds; REINVENT (Olivecrona et al. 2017) and MolGAN (De Cao and Kipf 2018) use text-based and graph-based representations, respectively. In RL, the atoms are attached to the main chain in order to attain more optimal property values. However, such a sequential method of generation is considered to be challenging to apply because the property values of a chemical substance can change significantly with or without local substructures, such as functional groups.

Another approach uses variational autoencoder (VAE) models; for example, JT-VAE using graph-based representation (Jin, Barzilay, and Jaakkola 2018) and ChemicalVAE using text-based representation (Gómez-Bombarelli et al. 2018). With VAEs, the chemical structure can be directly obtained by specifying one point in its latent space. Studies have shown that it is possible to optimize the property values by searching the latent space because of the continuity of the space. However, it is important to note that the previous studies required huge training datasets. ChemicalVAE and REINVENT were trained using supervised learning with 250,000 and 350,000 compound data extracted from the ZINC database, respectively. In most cases, when developing industrial chemical products, only a few hundred or thousand supervised training data, i.e., data with property values measured by experiments, are available for a single product family.

The purpose of this study is to propose a method for suggesting viable candidate compounds with improved properties even with a small amount of experimental data. In addi-

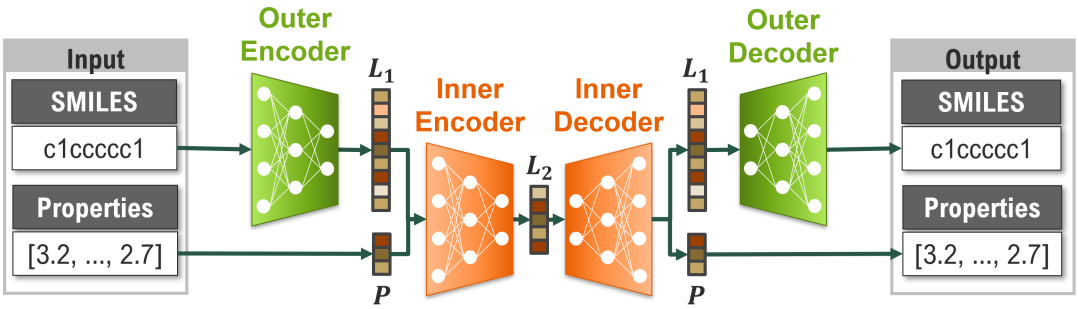

Figure 1: Diagram of the proposed nested variational autoencoders used for molecular generation.

tion, great attention should be paid to the transferability of the trained model because training deep generative models is costly in general. Hitachi, Ltd. provides such MI solutions, such as a cloud-based IT platform for non-experts (Osakabe, Asahara, and Morita 2020), to several material manufacturers. The proposed method is based on semi-supervised learning so that the trained model can be reused in different projects, which is advantageous to such business.

Related Works

Variational Autoencoder for Molecular Generation

Deep neural networks (DNNs) can be used to represent highly nonlinear relationships. They have outperformed other methods in various fields on regression and classification tasks. Recently, DNNs have been utilized to generate a novel sample similar to the training data, i.e., a deep generative model. The deep generative model assumes that the observed data x is generated from an unobserved latent variable z and aims to learn a transformation rule $p(x|z)$. One of the deep generative models is the variational autoencoder (VAE), which contains two neural networks, an encoder network, and a decoder network. The VAE assumes a continuous latent variable z to be a multidimensional Gaussian distribution. The encoder is trained to convert a data x to a latent variable z, and the decoder is simultaneously trained to convert z back to x. Gómez-Bombarelli et al. integrated this VAE framework into automatic molecular design, namely ChemicalVAE (Gómez-Bombarelli et al. 2018). Starting from a discrete molecular representation such as a SMILES string (Weininger 1988), the ChemicalVAE encoder converts this discrete representation x of a molecule into a real-valued continuous vector z, and its decoder converts z back to the discrete molecular representation x. With ChemicalVAE, the encoder utilizes a one-dimensional convolutional neural network (CNN) to convolve strings, and the decoder utilizes a recurrent neural network (RNN) to generate SMILES strings. By selecting a point in the latent space and passing it through the decoder, the corresponding SMILES can be obtained directly.

Two approaches can be used to generate a candidate molecule with the desired property values. One is to only use the trained VAE to generate the novel structure and predict the property values by calculating the descriptors with the conventional QSAR framework. The other approach is to build a new regression model with the latent variable and the property values. ChemicalVAE follows the second approach. To ensure the VAE generates effective candidate molecules that have improved property values, ChemicalVAE has an additional network, the predictor network, which estimates property values from the latent continuous vector z. Because of the continuity of the latent space, ChemicalVAE can automatically generate novel chemical structures by simple operations in the latent space, such as decoding random vectors, sampling the neighbors of known chemical structures, or interpolating between molecules. However, the continuity of the latent space is only guaranteed around the points of the known molecules, and the effectiveness of searching for novel candidates expected to have more optimal property values depends on the performance of the predictor network. In general, such a predictor requires a large amount of training data including both SMILES and corresponding property values. In most cases, it is difficult to prepare a large experimental dataset.

Semi-supervised Learning for VAE

A conditional VAE (CVAE) has been developed as an extension of VAE, which takes into account the objective variable y in the formulation of the latent variable z. Kingma et al. proposed a deep generative model called the M1+M2 model, based on CVAE architectures. The M1 model is trained with a large amount of unlabeled data which lacks information on the objective variable y, and then the M2 model is trained with the latent variable z by mixing a few labeled data and a large amount of unlabeled data. This model achieved 96% correctness in a handwriting image recognition task using MNIST where only 100 images out of 70,000 images were given the labels (correct answers). This approach can be understood as semi-supervised learning and is expected to be effective in generating molecules when limited by a small amount of experimental data. However, both the M1 and M2 models require a large amount of unlabeled data for training. When replacing the experimental data, the M2 model has to be trained again, which is not desirable in terms of computation time and diversion of the trained model.

Proposed Method

In this paper, we propose a generative model consisting of two nested VAEs by introducing semi-supervised learning

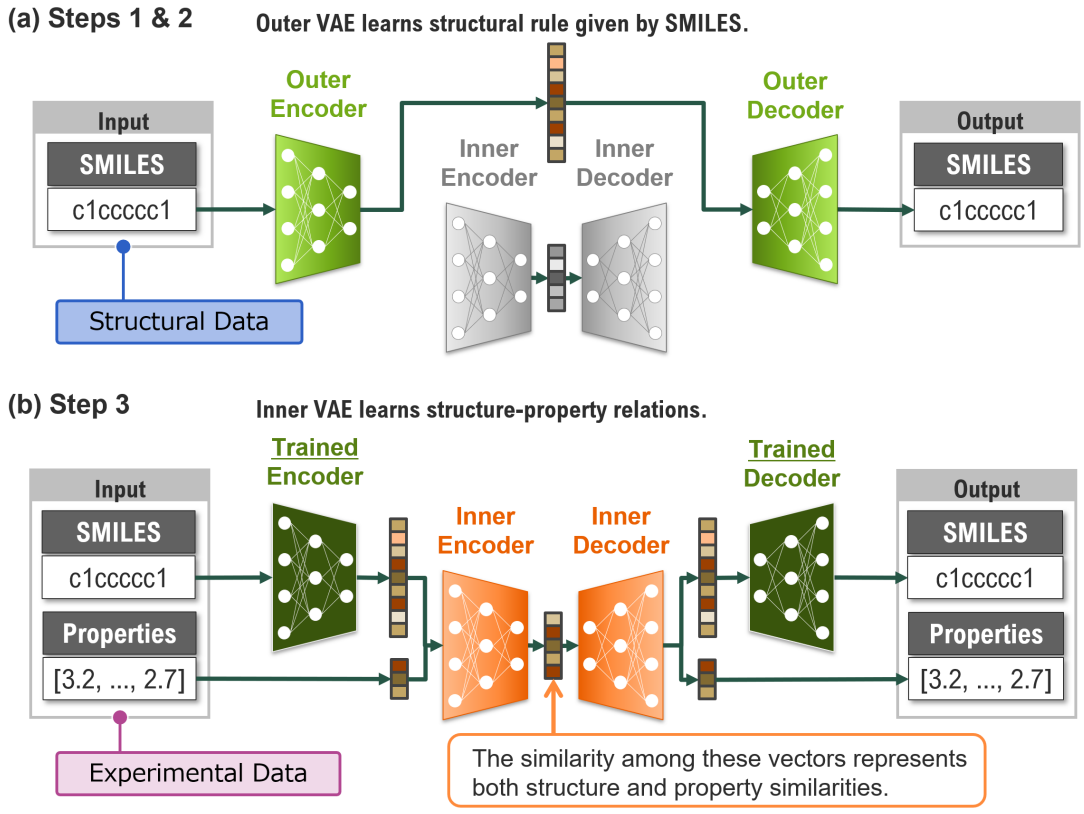

Figure 2: Diagram of learning steps; (a) steps 1 and 2 for training the outer VAE, and (b) step 3 for training the inner VAE.

similar to the M1+M2 model. To account for the diversion of the trained model, the two VAEs are trained independently on different datasets. Figure 1 shows the diagram of the proposed model, MatVAE. The first VAE (outer VAE) is trained to learn structure characteristics using a huge compound structure dataset without property values, such as an open dataset. The second VAE (inner VAE) is trained to learn the relation between the structure characteristics and the property values to be improved using an experimental dataset that includes both structure information and its property values. Unlike the M1+M2 model, the costly training with the huge dataset only needs to be done once for the outer VAE, and there is no need to replace the trained outer VAE if the experimental data changes. This model is expected to generate candidate compounds with improved property values by giving a vector similar to the existing top-level compounds paired with the desired property values.

Molecular Representations

SMILES (Weininger 1988) is a common format for representing molecules as a character sequence. With advances in natural language processing, this text-based format is widely used in MI applications such as predicting (Schwaller et al. 2018, 2019a; Coley et al. 2019; Bradshaw et al. 2019) and classifying (Schwaller et al. 2019b) chemical reactions. To make SMILES capable of inputting to VAE architectures, the SMILES strings are encoded into one-hot vectors made

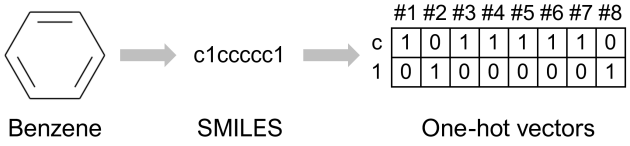

Figure 3: Example of SMILES representation and one-hot vectors for benzene. For simplicity, only two characters are shown in the one-hot encoding. In practice, one-hot vectors forms $M \times N$ matrix, where M is the number of SMILES symbols and N is the maximum length of SMILES strings.

up of $M \times N$ matrix, where M is the number of SMILES symbols and N is the maximum length of SMILES strings. In our experiment, M is 101 and N is 90. The one-hot vector indicates the presence and absence of each symbol within a sequence, as illustrated in Fig. 3. InChI (Heller et al. 2013) is another common representation of molecules, but Gómez-Bombarelli et al. reported that it is less effective than SMILES with VAE architectures (Gómez-Bombarelli et al. 2018). Note that SELFIES (Krenn et al. 2020) is another text-based representation based on the combination of strings and graph expressions of molecules. SELFIES was found to be highly robust against mutations in sequences and outperformed other representations (including SMILES strings) in terms of diversity, validity, and reconstruction accuracy when applied to sequence-based VAEs. SELF-

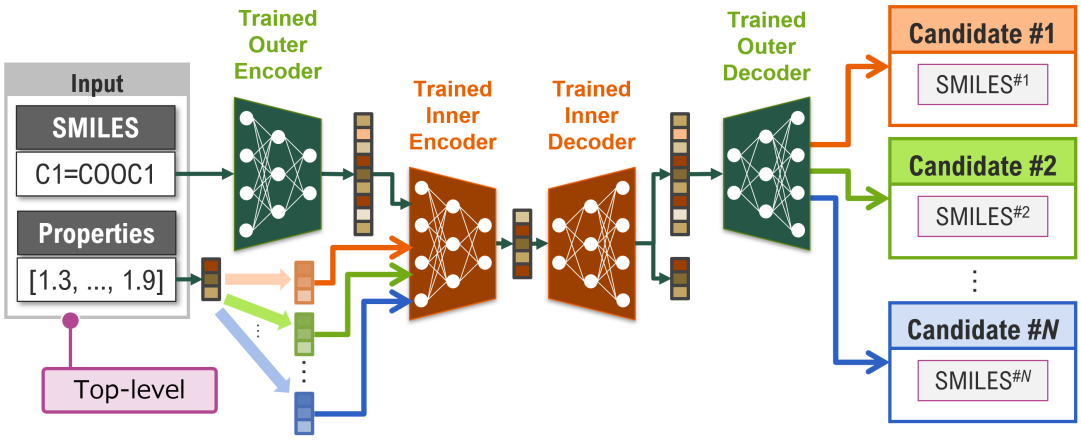

Figure 4: Diagram of the generation phase for MatVAE.

IES is compatible with SMILES and can be handled with one-hot encoding similarly to SMILES. We have confirmed that SELFIES improves the validity of generated molecular strings with MatVAE, but in this paper SMILES is used as an input format to clarify whether the proposed nested VAEs are more effective than the conventional single VAE model (Gómez-Bombarelli et al. 2018) for generating candidate compounds.

Outer VAE

The purpose of the outer VAE is to prepare a general generative model for creating a one-hot vector that can be transformed into corresponding valid SMILES strings. Since the outer VAE does not need property value information, the SMILES training dataset can be easily curated from any open dataset such as ZINC (Irwin et al. 2012). All of the SMILES are converted into one-hot vectors before they are input to the outer VAE.

The structure of the outer VAE network is based on ChemicalVAE (Gómez-Bombarelli et al. 2018) as follows: the outer encoder utilizes the three one-dimensional convolutional layers to convolve strings, where the filter sizes are 9, 9, 10 and the convolution kernels are 9, 9, 11, followed by one fully connected layer of dimension $L_1 = 196$, where L_1 is the size of the latent space of outer VAE. The decoder utilizes three layers of gated recurrent unit (GRU) networks (Chung et al. 2014) with a hidden dimension of 488. The final layer of the decoder outputs a probability distribution over all possible symbols at each position in a SMILES string. As a result, the same point in latent space can be decoded into a different SMILES string, and the generated string may be invalid. Similarly to (Gómez-Bombarelli et al. 2018), the GRU layer is updated to improve its performance by using additional input (Williams and Zipser 1989).

Inner VAE

The purpose of the inner VAE is to learn the structure–property relations. By using the latent vector of the trained outer VAE instead of one-hot vectors to express molecular structure, the inner VAE should be able to extract such relations more easily because the input dimension is reduced.

The input of the inner VAE is a vector concatenated with a latent vector of the outer VAE and a vector consisting of property values. Therefore, the input size can be $L_1 + P$, where P is the number of the target properties to be improved or optimized, as shown in Fig. 2 (b).

The structure of the inner VAE network is as follows: both the inner encoder and decoder comprise multiple fully connected layers with the latent space of width L_2. Therefore, the number of nodes in each layer is automatically determined to be equal to the ratio between $L_1 + P$ and L_2. In this report, the latent space dimension L_2 is defined to be 128, and the inner encoder and decoder are both defined to have three layers.

Learning and Generating Procedures

The inner and the outer VAEs are trained independently. Again, the required datasets for training the two VAEs are (1) the structure dataset containing only SMILES and curated from open datasets, and (2) the experimental dataset containing both SMILES and experimental property values.

Figure 2 shows the following three steps which make up the learning procedure:

Step 1 Train outer VAE with structure dataset

Step 2 Train outer VAE with SMILES in experimental dataset

Step 3 Train inner VAE with experimental dataset

At a glance, Step 2 appears to be redundant. However, it is necessary to ensure the continuity in the latent spaces even around compounds in the experimental dataset. Unlike the M1+M2 model, the inner VAE is only trained on the labeled data with the experimental dataset. Hence, if the outer VAE does not learn those compounds, the two VAEs cannot work together in the generation phase. In fact, the generation of candidate compounds with the nested VAEs without Step 2 resulted in almost no valid SMILES generated.

Figure 4 illustrates the generation phase. To generate the candidate compounds expected to have improved proper-

Role	Size	Components	SA Score Range
Structure	500k	SMILES	–
Experimental	1k	SMILES, SA Score	$s_i \geq 2.05$
Top-level	0.1k	SMILES	$2.05 \geq s_i \geq 2.00$

Table 1: Overview of the training datasets in the experiment.

ties, we use the vector with the width $L_1 + P$ consisting of the latent vector similar to the existing top-level compounds paired with the desired property values. Candidate compounds can then be generated more efficiently by comprehensively combining multiple values for a single latent vector within a range close to the target property value.

Experiment

Setup

Candidate compounds generated by MatVAE are not guaranteed to be synthesizable even if they are valid according to the SMILES grammar. In this experiment, the synthetic accessibility (SA) score (Ertl and Schuffenhauer 2009) is used instead of the actual chemical properties so that the true value for the unknown SMILES can be calculated without conducting actual experiments. The SA score is an index of ease of fabrication and is uniquely determined for any valid SMILES.

One of the objectives of MatVAE is to generate candidate compounds that have improved property values. In other words, the model is expected to generate SMILES with property values outside the range of the given experimental dataset. To evaluate MatVAE, we deliberately limited the range of SA scores included in the experimental training dataset, as shown in Table 1. The experimental dataset is limited to include compounds in which s_i is larger than 2.05, where s_i denotes the SA score of the i-th compounds in the dataset. This means that the inner VAE cannot learn the structure–property relation in the range below 2.05. One hundred compounds with $2.05 \geq s_i \geq 2.00$ were curated for a top-level compounds dataset used as the VAE input in the generation phase. Note that all of the compounds in this experiment were curated from the ZINC database.

For the evaluation, the proposed model generated candidate compounds repeatedly until valid SMILES are generated, and the number of the generated compounds with SA scores below 2.0 was counted. If MatVAE is able to generate such compounds, it would be confirmed that the model successfully suggests unknown compounds that exceed the existing top-level compounds.

Results

Figure 5 shows the number of valid, optimal compounds from the 100 suggested valid candidates, and the red line indicates the performance of MatVAE. Even when the experimental dataset is small, MatVAE successfully generates valid SMILES with an improved SA score, though the ratio is not high. The blue line indicates the performance of the conventional method, ChemicalVAE, which is considerably lower than that of the original results, though all parameters and hyperparameters are the same as those of the original model. This is because ChemicalVAE is not intended to be trained on such a small amount of experimental data. In fact, when the number of experimental data is increased to at least 5000, ChemicalVAE outperforms MatVAE. Thus, for a small amount of experimental data, MatVAE can produce more than five times as many optimal compounds as ChemicalVAE.

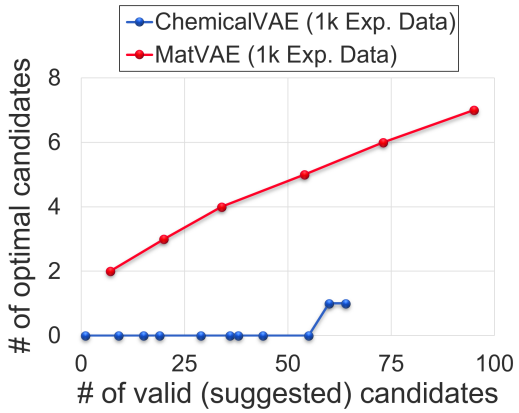

Figure 5: The number of optimal and valid candidates generated by the conventional ChemicalVAE and the proposed MatVAE.

Conclusion

We have proposed a new deep generative model, MatVAE, for producing molecules with optimal property values. The model consists of two variational autoencoders which are trained independently on different datasets. The first (outer) VAE aims to capture the structure characteristics according to the grammar of a text-based representation of molecules such as SMILES. Even if the type of target properties changes, this VAE does not need to be replaced, making it transferable to other applications. The second (inner) VAE can learn structure–property relations directly using the experimental dataset because the first VAE extracts structural features of molecules by reducing the dimensions of the one-hot vectors converted from SMILES. Therefore, it can generate new molecule representations by inputting existing top-level compounds structure data paired with the desired property values. Compared with the common optimization methodology for VAE, which utilizes gradient-based search in its latent space, the proposed search method is more straightforward and user-friendly.

There are a number of possible improvements that can be made to MatVAE. In this study, we used text-based molecular encoding and a GRU decoder. Such architecture may make it unnecessarily difficult to produce valid SMILES strings. It is possible to use a graph-based representation instead of text-based, or to change the sequence-based VAE to graph-based. However, even without changing the network configuration, converting SMILES to one-hot vectors via SELFIES representation could prevent the generation of invalid SMILES. We have already verified the effectiveness of this approach in other experiments.

Generated compounds are likely to have similar substructures, such as carbon chains, because VAEs tend to produce frequent patterns in the training dataset. In fact, MatVAE is more likely to produce compounds with a simple structure and a low SA score than compounds with a complex structure and a high SA score. To generate useful and novel candidate compounds while maintaining structural diversity, it will be necessary to modify the loss function in the future to include restrictions.

References

Bradshaw, J.; Paige, B.; Kusner, M. J.; Benevolentai, M. H. S. S.; and Miguel Hernández-Lobato, J. 2019. A Model to Search for Synthesizable Molecules. Technical report. URL https://github.com/.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling URL http://arxiv.org/abs/1412.3555.

Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; and Jensen, K. F. 2019. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical Science* 10(2): 370–377. ISSN 20416539. doi:10.1039/c8sc04228d. URL https://pubs.rsc.org/en/content/articlehtml/2019/sc/c8sc04228dhttps://pubs.rsc.org/en/content/articlelanding/2019/sc/c8sc04228d.

Curtarolo, S.; Hart, G. L.; Nardelli, M. B.; Mingo, N.; Sanvito, S.; and Levy, O. 2013. The high-throughput highway to computational materials design. *Nature materials* 12(3): 191.

De Cao, N.; and Kipf, T. 2018. MolGAN: An implicit generative model for small molecular graphs URL http://arxiv.org/abs/1805.11973.

Ertl, P.; and Schuffenhauer, A. 2009. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics* 1(1): 8. ISSN 1758-2946. doi:10.1186/1758-2946-1-8. URL https://jcheminf.biomedcentral.com/articles/10.1186/1758-2946-1-8.

Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; and Aspuru-Guzik, A. 2018. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* 4(2): 268–276. ISSN 2374-7943. doi:10.1021/acscentsci.7b00572. URL https://pubs.acs.org/doi/10.1021/acscentsci.7b00572.

Heller, S.; McNaught, A.; Stein, S.; Tchekhovskoi, D.; and Pletnev, I. 2013. InChI - The worldwide chemical structure identifier standard. doi:10.1186/1758-2946-5-7. URL https://jcheminf.biomedcentral.com/articles/10.1186/1758-2946-5-7.

Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; and Coleman, R. G. 2012. ZINC: A Free Tool to Discover Chemistry for Biology. *Journal of Chemical Information and Modeling* 52(7): 1757–1768. ISSN 1549-9596. doi:10.1021/ci3001277. URL https://pubs.acs.org/doi/10.1021/ci3001277.

Jin, W.; Barzilay, R.; and Jaakkola, T. 2018. Junction tree variational autoencoder for molecular graph generation. *35th International Conference on Machine Learning, ICML 2018* 5: 3632–3648.

Krenn, M.; Hase, F.; Nigam, A.; Friederich, P.; and Aspuru-Guzik, A. 2020. Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology* doi:10.1088/2632-2153/aba947. URL https://creativecommons.org/licences/by/3.0.

Olivecrona, M.; Blaschke, T.; Engkvist, O.; and Chen, H. 2017. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics* 9(1). ISSN 17582946. doi:10.1186/s13321-017-0235-x.

Osakabe, Y.; Asahara, A.; and Morita, H. 2020. Hitachi Materials Informatics Analytics Platform Assisting Rapid Development. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1)*.

Schwaller, P.; Gaudin, T.; Lányi, D.; Bekas, C.; and Laino, T. 2018. "Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical Science* 9(28): 6091–6098. ISSN 20416539. doi:10.1039/c8sc02339e. URL https://pubs.rsc.org/en/content/articlehtml/2018/sc/c8sc02339ehttps://pubs.rsc.org/en/content/articlelanding/2018/sc/c8sc02339e.

Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; and Lee, A. A. 2019a. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Central Science* 5(9): 1572–1583. ISSN 23747951. doi:10.1021/acscentsci.9b00576. URL http://pubs.acs.org/journal/acscii.

Schwaller, P.; Probst, D.; Vaucher, A. C.; Nair, V. H.; Laino, T.; and Reymond, J.-L. 2019b. Data-Driven Chemical Reaction Classification, Fingerprinting and Clustering using Attention-Based Neural Networks doi:10.7892/boris.141739. URL https://doi.org/10.7892/boris.141739.

Weininger, D. 1988. SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules. *Journal of Chemical Information and Computer Sciences* 28(1): 31–36. ISSN 00952338. doi:10.1021/ci00057a005.

Williams, R. J.; and Zipser, D. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation* 1(2): 270–280. ISSN 0899-7667. doi:10.1162/neco.1989.1.2.270. URL https://www.mitpressjournals.org/doix/abs/10.1162/neco.1989.1.2.270.