Transmol: Repurposing language model for molecular generation

Rustam Zhumagambetov,[†] Vsevolod A. Peshkov,[‡] and Siamac Fazli[*,†]

[†]*Department of Computer Science, Nazarbayev University, Nur-Sultan*

[‡]*Department of Chemistry, Nazarbayev University, Nur-Sultan*

E-mail: {rustam.zhumagambetov,vsevold.peshkov,siamac.fazli}@nu.edu.kz

Abstract

Recent advances in convolutional neural networks have inspired the application of deep learning to other disciplines. Even though image processing and natural language processing have turned out to be the most successful, there are many other areas that have benefited, like computational chemistry in general and drug design in particular. From 2018 the scientific community has seen a surge of methodologies related to the generation of diverse molecular libraries using machine learning. However, no algorithm used an attention mechanisms for *de novo* molecular generation. Here we employ a variant of transformers, a recent NLP architecture, for this purpose. We have achieved a statistically significant increase in some of the core metrics of the MOSES benchmark. Furthermore, a novel way of generating libraries fusing two molecules as seeds has been described.

Introduction

Chemistry is frequently referred to as a "central science" for its key role in advancing technological progress and human well-being through the design and synthesis of novel molecules

and materials for energy, environmental, and biomedical applications.

Medicinal chemistry is a highly interdisciplinary field of science that deals with the design, chemical synthesis, and mechanism of action of biologically active molecules as well as their development into marketed pharmaceutical agents (i.e. drugs). The creation of new drugs is an incredibly hard and arduous process. One of the key reasons for that comes from the fact that the 'chemical space' of all possible molecules is extremely large and intractable. Even though it is estimated that the chemical space of molecules with pharmacological properties is of the range of $10^{23} - 10^{60}$ compounds,[1] this order of magnitude leaves the work of finding new drugs out of reach of manual labor.

In general, medicinal chemists need to determine molecules that are active and selective towards specific biological targets to cure a particular disease while keeping the risks of negative side effects at a minimal level. As the number of molecules that require testing to identify an ideal drug candidate constantly increases, it raises the overall cost of the drug discovery process. Therefore, the need for algorithms that are able to narrow down and optimize these efforts has recently emerged. Specifically, the computer algorithms can assist with creating new virtual molecules as well as with performing conformational analysis and molecular docking to determine the affinity of novel and known molecules towards specific biological targets.

With respect to molecular generation, the conventional non-neural algorithms heavily rely on external expert knowledge to construct candidate molecules. In this context, expert knowledge may consist of molecular compounds/fragments that could be "mixed and matched" together to produce a set of potential molecules.[2] However, the resulting molecules might be difficult to synthesize. Another type of expert knowledge can be then added: known chemical reactions.[3,4] It is possible to constrain this "mix and match" procedure using a known system of rules to ensure that any molecule that was produced can be synthesized. However, it is known that such systems can have some limitations.[5] Besides, relying on external knowledge may result in restricting access to unknown and/or not yet populated

regions of chemical space.

An alternative approach to this problem would be neural algorithms that are inherently data-driven. This means that such algorithms do not rely on expert knowledge and hence derive insights from the data itself. Such approaches can be applied in supervised and unsupervised fashions. Supervised algorithms use artificial neural networks for the prediction of molecular properties[6] or reaction outputs.[7] Most unsupervised algorithms are aimed at molecular generation and drug design.[8–21]

Researchers have used a number of techniques for a molecular generation: generative adversarial networks, variational autoencoders, and recurrent neural networks. However, currently, the use of attention mechanisms for *de novo* drug design, to our knowledge, is unexplored. One of our goals is to fill this gap and investigate the applicability of attention to molecular generation.

We set a goal to develop the algorithm that would allow the generation of the diverse focused libraries utilizing one, or two seed molecules that will guide the generation of *de novo* molecules. It outperforms existing artificial neural network algorithms in some core MOSES metrics, a benchmark introduced for the comparison of generative algorithms:[22] internal diversity (IntDiv_1), and similarity to a nearest neighbor (SNN). The resulting algorithm is incorporated into the cheml.io[23] website and available for the generation of molecules on demand.

Dataset

In this paper, we have used a MOSES benchmark and the dataset it provided. It consists of three datasets: training, testing, and testing scaffolds, containing 1.6M, 176k, and 176k respectively.

The first dataset is recommended for use for the training of the model. The model will learn to interpolate between each molecule and construct a latent space. The latent space

further will act as a proxy distribution for molecules. So it would be possible to sample new molecules from it.

The testing dataset consists of molecules that are not present in the training dataset. It is used to assess how effectively the model is generalizable: whether the architecture of the model can be applied to other datasets.

The scaffold testing dataset consists of scaffolds that are not present in the training and testing datasets. Scaffolds are small fragments of molecules that can describe a set of compounds, where it is present. The scaffold testing is used to check if the model can generate new scaffolds, unique molecular features, or whether the model just reuses the parts of the previously seen molecules to generate new ones. It is more desirable for the model to generate new scaffolds as it would mean that it has learning capabilities.

Molecular representation

When beginning discussion of automating molecular search a natural question is how molecules, a physical collection of atoms that are arranged in 3D space, can be represented.

In the 1980s simplified molecular input line-entry system (SMILES) specification had emerged, aimed to create a molecular encoding that is computationally efficient to use and is human readable.[24] The original encoding is based on the 2D molecular graphs. Intended application areas are fast and compact information retrieval and storage. With the rise of machine learning algorithms in chemistry, SMILES have been widely adopted by researchers for chemical space modeling tasks.

Data augmentation

To improve the validity of the algorithm we have used data augmentation through SMILES enumeration as was used in work Arús-Pous et al. [25] A molecule can be mapped to the unique canonical SMILES string, however non-unique SMILES strings can be also produced depending on the starting point where the algorithm will begin translation. Such data

augmentation has been reported to improve the generalization of the latent space (increase the diversity of the output molecules).

Method

For this work, we have employed a vanilla transformer model from the work by Vaswani et al.[26] A vanilla transformer consists of two parts: encoder and decoder. An encoder (see left dashed block of Figure 1) maps input to the latent representation z. A decoder (see right dashed block of Figure 1), accepts z as input and produces one symbol at a time. The model is auto-regressive, i.e to produce a new symbol it requires previous output as an additional input.

The notable attribute of this architecture is the use of attention mechanisms throughout the whole model. While models before transformers have been using attention only as an auxiliary layer, having some kind of recurrent neural networks (RNN) like gated recurrent unit (GRU) or long short-term memory (LSTM), or convolutional neural network (CNN), the transformer consists primarily of attention layers.

Attention mechanism can looked at as function of query Q, key K and value V, where output is a matrix product of Q, K, V using function:

$$\text{Scaled dot-product Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

is used to identify the relevant parts of the input in respect to the input, self-attention. It allows disregarding less important parts of the query and filter noise. The most important part is that attention mechanisms are differentiable, hence can be learned from data. See Figure 2 for the illustration of the scaled dot-product attention layer. The multi-head attention layer consists of h instances of scaled dot-product attention layers that are then concatenated and passed to the dense layer.

Parameters of the original paper have been used, such as number of stacked encoder and

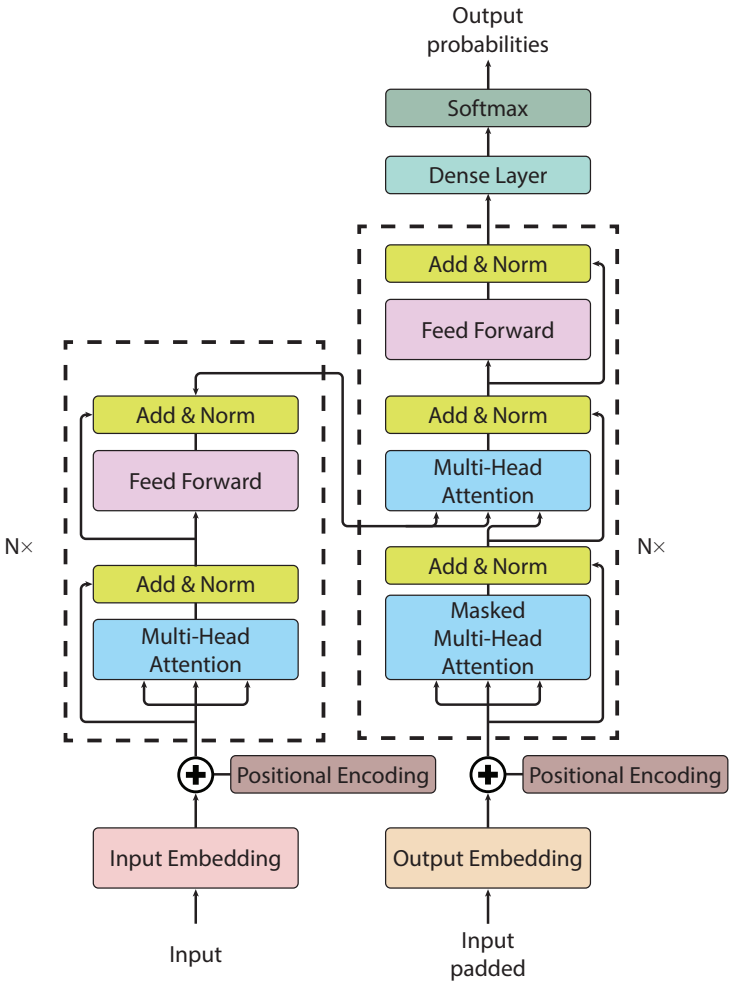


Figure 1: A vanilla transformer architecture

decoder layers $N = 6$, all sublayers produce output of $d_{\text{model}} = 512$, with dimensionality of inner feed-forward layer being d_{ff}, number of attention heads $h = 6$, and dropout $d = 0.1$.

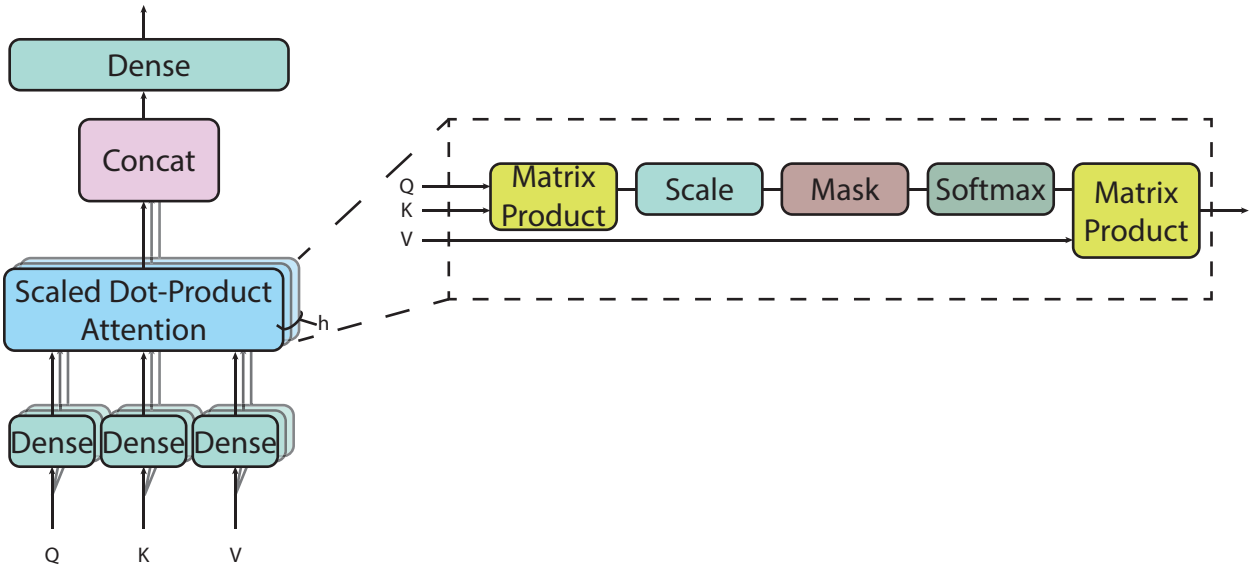


Figure 2: Multi-head attention layer

Sampling from the latent space

To sample the model, a seed SMILES string is needed to provide a context for the decoder. Then the decoding process is started by supplying a special starting symbol. After that the decoder provides an output and a first symbol is generated. To get the next symbol the previous characters are provided to the decoder. The decoding process stops when the decoder either outputs a special terminal symbol or exceeds the maximum length. There are several techniques available that specify how the output of the decoder is converted to the SMILES character such as a simple greedy search or a beam search.

Greedy search

As the decoder provides output probabilities the naive approach would be using a greedy algorithm and picking the symbol with the highest probability. However, it is not optimal as picking the most probable symbol on each step does not guarantee that the final resulting

string would have the highest conditional probability. Moreover, unless stochastic sampling is used (when probability vector is used as a basis for the distribution and then sampled), the result of the greedy search is deterministic and corresponds to the "reconstruction accuracy" based on our training procedure.

Beam search

To improve upon the greedy search a beam search has been used. The beam search is an improved greedy search. While the greedy search picks only one symbol at a time, the beam search picks N most probable. Figure 3 illustrates the beam search with beam width $N = 3$; the stroke width indicates the probability. To guide the selection process of beam search we have used the following reward function:

$$\frac{\sum_{char \in vocab} P(s|\text{previous output})}{(1 + |\text{previous output}|)^{\alpha}}$$

where *char* is a possible symbol for beam search to pick, *vocab* is a set of all possible characters, the *previous output* is an ordered list of symbols picked by beam search prior to picking current one, α is a parameter of beam search that regulates the length of the string, low α discourages long strings, high α encourages.

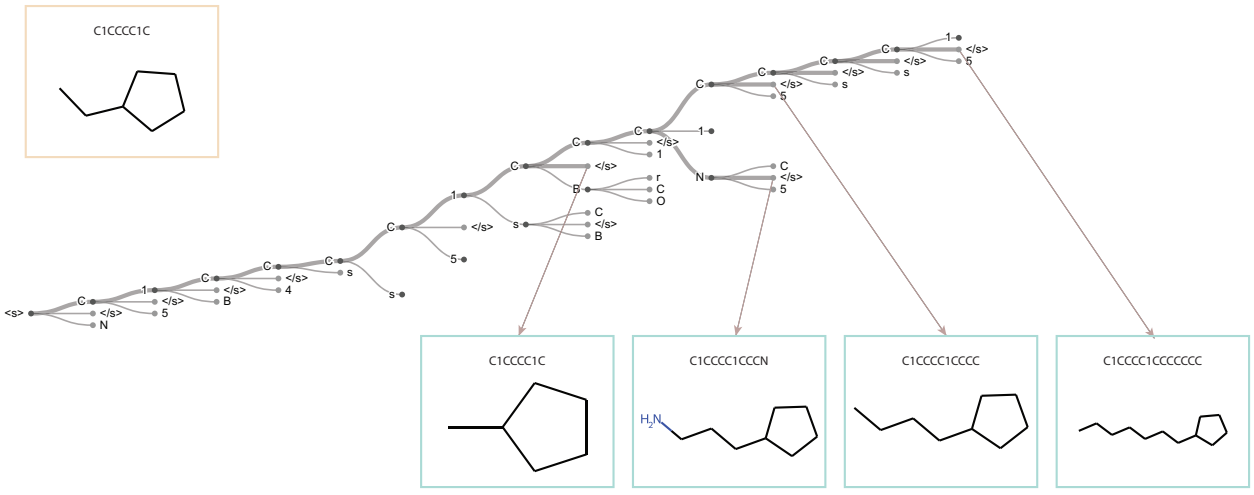


Figure 3: Overview of the beam search with a beam width of N=3

Injecting variability into model

To explore the molecules that are located near the seed molecule in the latent space, we have used two techniques that allow sample from the seed cluster: addition of Gaussian noise to the z and use of temperature.

Gaussian noise

To increase the variability of the model we are adding the Gaussian noise with a mean of μ and standard deviation σ to latent vector z before it is fed to the decoder.

Temperature

Another technique to improve variability is to apply temperature to the output vector right before applying the softmax function. Temperature T is a value from 0 to ∞. As $T \to \infty$ all characters have the same probability of being the next symbol. For $T \to 0$ the most probable symbol has a higher probability of being selected. Figure 4 demonstrates how the application of temperature smoothes the original distribution. The resulting smoothed distribution increases the variability of sampling.

Results

In this section, we describe major results that were obtained during the experiments. It starts with the generation of a focused library with a single seed molecule which is followed by the description of the generation of a focused library using two seed molecule. See Figure 5 for the graphical overview of the process. The top figure illustrates sampling from the latent space using only one seed molecule; bottom figure illustrates the similar process for two seed molecules.

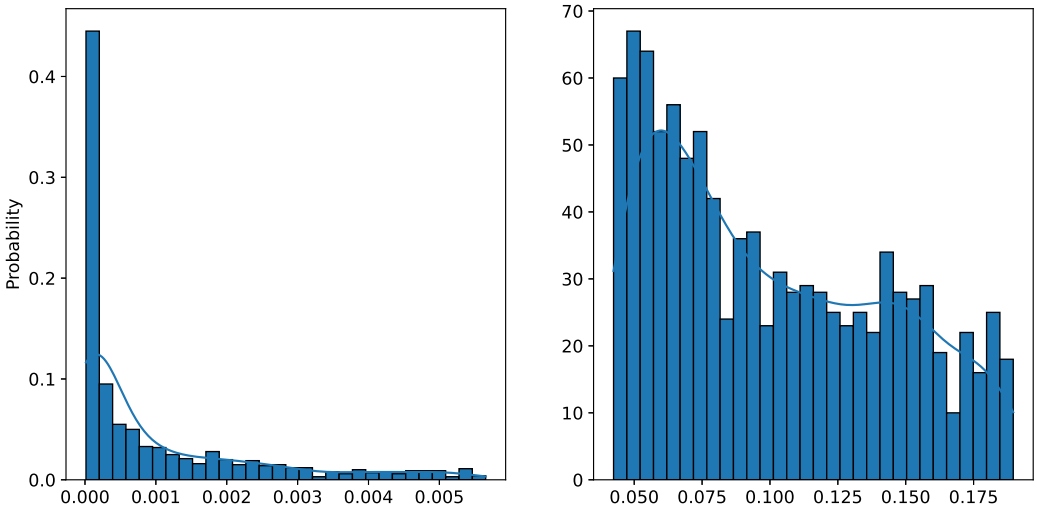


Figure 4: Impact of temperature on distribution

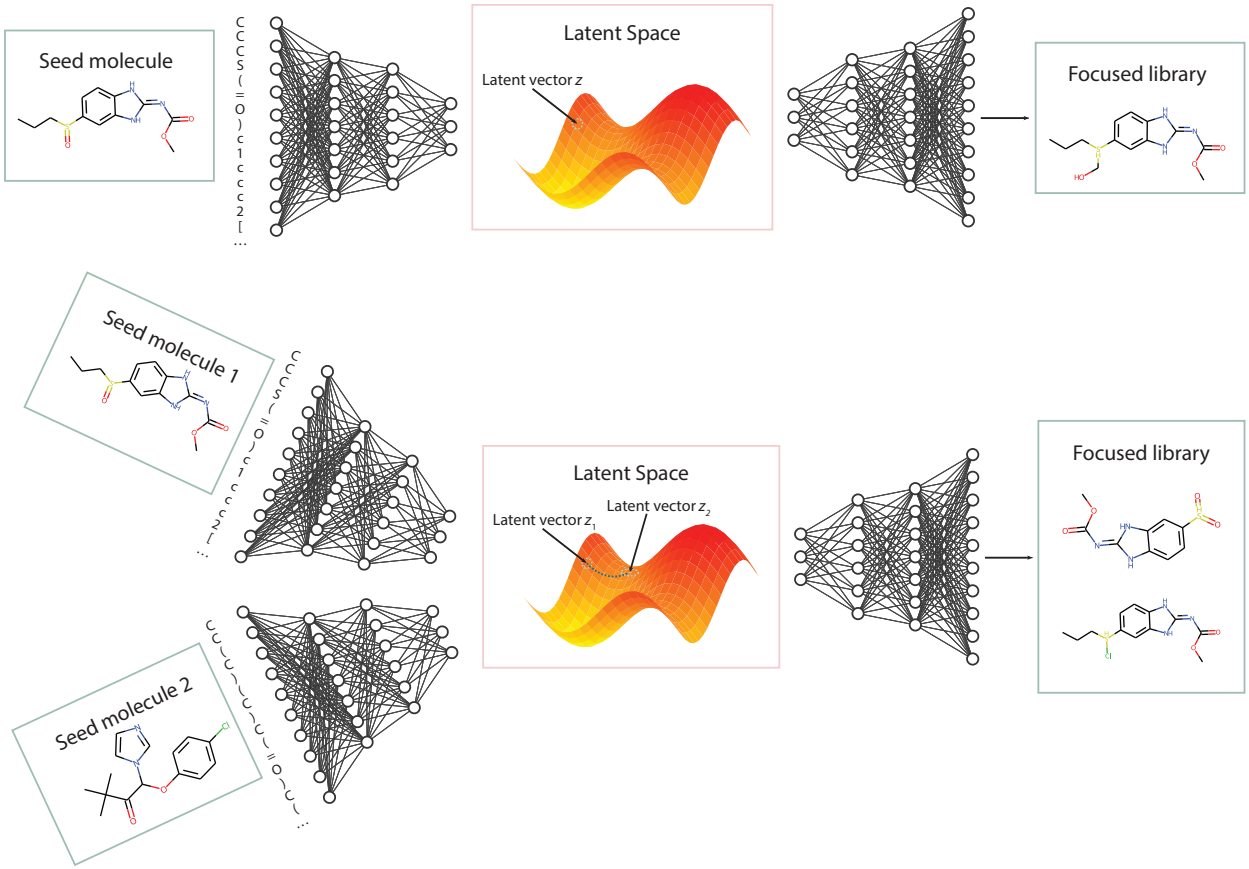


Figure 5: The general pipeline of the sampling process

Creating focused library with seed molecule

In this subsection, we discuss the optimization procedure for the sampling hyperparameters as well as present the results of the MOSES benchmark[22] in relation to our method and previous ones.

MOSES metrics

Several metrics are provided by the MOSES benchmark. Uniqueness, validity, and internal diversity are among the most important ones.

Uniqueness shows the proportion of generated molecules that are within the training dataset. Validity describes the proportion of generated molecules that are chemically sound. Internal diversity measures whether the model samples from the same region of chemical space, producing molecules that are valid and unique but differ in a single atom; hence, are useless.

Table 1 demonstrates that Transmol has demonstrated the greatest diversity across all baselines. It can be also observed that among neural algorithms Transmol demonstrates the greatest proportion of novel molecules, that are not present in the training dataset. Table 2 demonstrates performance metrics: Fréchet ChemNet Distance (FCD), Similarity to a nearest neighbor (SNN), Fragment similarity (Frag), and Scaffold similarity (Scaff). In the SNN metric Transmol demonstrates the advantage over the baselines models.

Figure 6 demonstrates the Wasserstein-1 distance between generated molecules and the test set (in brackets). It shows that the Transmol is not as close to the testing set distribution as other neural algorithms, but it is not as far from it as simpler, combinatorial baselines. Kernel density estimation is visualized using a distribution plot.

Adjusting beam search

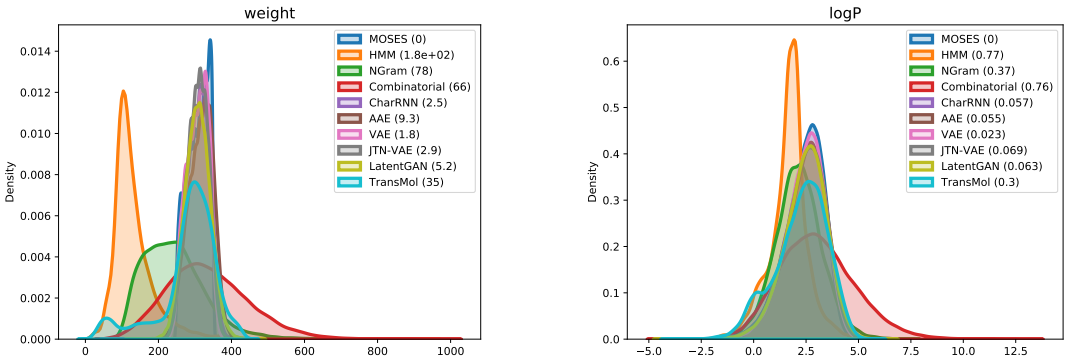
To find the optimal parameters for the beam search a grid search has been conducted. Figure 7 shows the dependency between beam width and topk(number of generated molecules).

Table 1: Performance metrics for baseline models: fraction of valid molecules, fraction of unique molecules from 1,000 and 10,000 molecules, internal diversity, fraction of molecules passing filters (MCF, PAINS, ring sizes, charge, atom types), and novelty. Reported (mean ± std) over three independent model initializations.

Model	Valid (↑)	Unique@1k (↑)	Unique@10k (↑)	IntDiv (↑)	IntDiv2 (↑)	Filters (↑)	Novelty (↑)
Train	*1*	*1*	*1*	*0.8567*	*0.8508*	*1*	*1*
HMM	0.076±0.0322	0.623±0.1224	0.5671±0.1424	0.8466±0.0403	0.8104±0.0507	0.9024±0.0489	**0.9994±0.001**
NGram	0.2376±0.0025	0.974±0.0108	0.9217±0.0019	0.8738±0.0002	0.8644±0.0002	0.9582±0.001	0.9694±0.001
Combinatorial	**1.0±0.0**	0.9983±0.0015	0.9909±0.0009	0.8732±0.0002	**0.8666±0.0002**	0.9557±0.0018	0.9878±0.0008
CharRNN	0.9748±0.0264	**1.0±0.0**	0.9994±0.0003	0.8562±0.0005	0.8503±0.0005	0.9943±0.0034	0.8419±0.0509
AAE	0.9368±0.0341	**1.0±0.0**	0.9973±0.002	0.8557±0.0031	0.8499±0.003	0.996±0.0006	0.7931±0.0285
VAE	0.9767±0.0012	**1.0±0.0**	0.9984±0.0005	0.8558±0.0004	0.8498±0.0004	**0.997±0.0002**	0.6949±0.0069
JTN-VAE	**1.0±0.0**	**1.0±0.0**	**0.9996±0.0003**	0.8551±0.0034	0.8493±0.0035	0.976±0.0016	0.9143±0.0058
LatentGAN	0.8966±0.0029	**1.0±0.0**	0.9968±0.0002	0.8565±0.0007	0.8505±0.0006	0.9735±0.0006	0.9498±0.0006
Transmol	0.0682±0.0022	0.9420±0.0026		**0.8779±0.0016**	0.8651±0.0017	0.8504±0.0098	0.9821±0.0020

Table 2: Performance metrics for baseline models: Fréchet ChemNet Distance (FCD), Similarity to a nearest neighbor (SNN), Fragment similarity (Frag), and Scaffold similarity (Scaff); Reported (mean ± std) over three independent model initializations. Results for random test set (Test) and scaffold split test set (TestSF)

Model	FCD (↓)		SNN (↑)		Frag (↑)		Scaf (↑)	
	Test	TestSF	Test	TestSF	Test	TestSF	Test	TestSF
Train	*0.008*	*0.4755*	*0.6419*	*0.5859*	*1*	*0.9986*	*0.9907*	*0*
HMM	24.4661±2.5251	25.4312±2.5599	0.3876±0.0107	0.3795±0.0107	0.5754±0.1224	0.5681±0.1218	0.2065±0.0481	0.049±0.018
NGram	5.5069±0.1027	6.2306±0.0966	0.5209±0.001	0.4997±0.0005	0.9846±0.0012	0.9815±0.0012	0.5302±0.0163	0.0977±0.0142
Combinatorial	4.2375±0.037	4.5113±0.0274	0.4514±0.0003	0.4388±0.0002	0.9912±0.0004	0.9904±0.0003	0.4445±0.0056	0.0865±0.0027
CharRNN	**0.0732±0.0247**	**0.5204±0.0379**	0.6015±0.0206	0.5649±0.0142	**0.9998±0.0002**	0.9983±0.0003	0.9242±0.0058	**0.1101±0.0081**
AAE	0.5555±0.2033	1.0572±0.2375	0.6081±0.0043	0.5677±0.0045	0.991±0.0051	0.9905±0.0039	0.9022±0.0375	0.0789±0.009
VAE	0.099±0.0125	0.567±0.0338	0.6257±0.0005	**0.5783±0.0008**	0.9994±0.0001	**0.9984±0.0003**	**0.9386±0.0021**	0.0588±0.0095
JTN-VAE	0.3954±0.0234	0.9382±0.0531	0.5477±0.0076	0.5194±0.007	0.9965±0.0003	0.9947±0.0002	0.8964±0.0039	0.1009±0.0105
LatentGAN	0.2968±0.0087	0.8281±0.0117	0.5371±0.0004	0.5132±0.0002	0.9986±0.0004	0.9972±0.0007	0.8867±0.0009	0.1072±0.0098
Transmol	6.3070±0.04197	7.1923±0.0833	**0.6290±0.0048**	0.4662±0.0038	0.9432±0.0034	0.9374±0.0033	0.5224±0.0471	0.0095±0.0064



(a) Weight (b) logP

(c) QED (d) SA

Figure 6: Plots of Wasserstein-1 distance between distributions of molecules in the generated and test sets

The chart shows that for small numbers of topk the fraction of valid molecules is high.

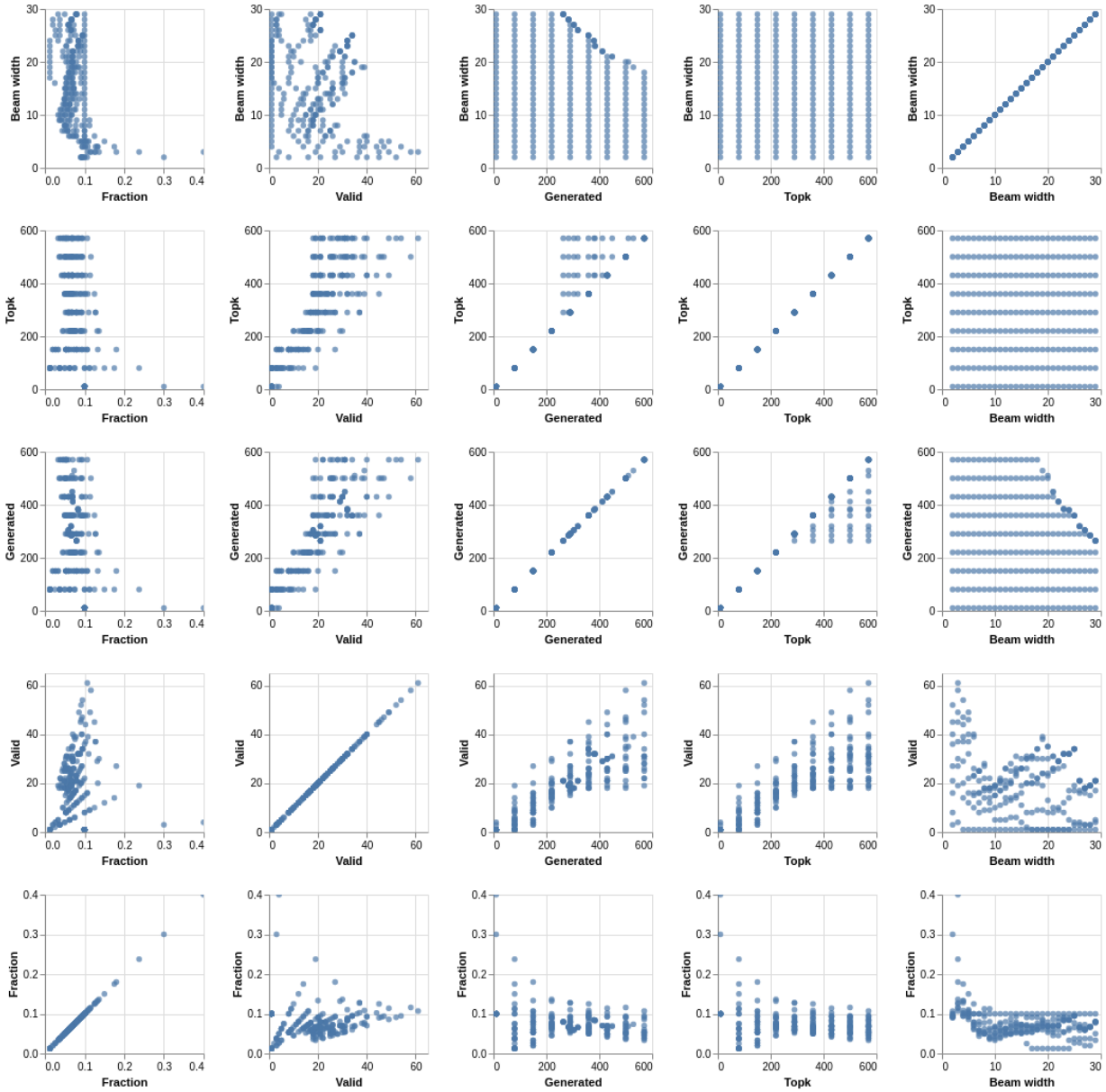


Figure 7: The scatter plot of the grid search with the following parameters: beam width, number of generation requests, actual number of generated smiles, number of valid molecules, and fraction of valid molecules

Exploring chemical space using two seed molecules

Since no known benchmark involves multiseed sampling in this subsection we describe a procedure of the encoder verification, and the procedure of adjusting the weights. Figure 8

illustrates molecular sampling from the latent distribution using two seed molecules. The resulting generated library demonstrates a diversity of structural features that would be unattainable through simple fragment substitution.

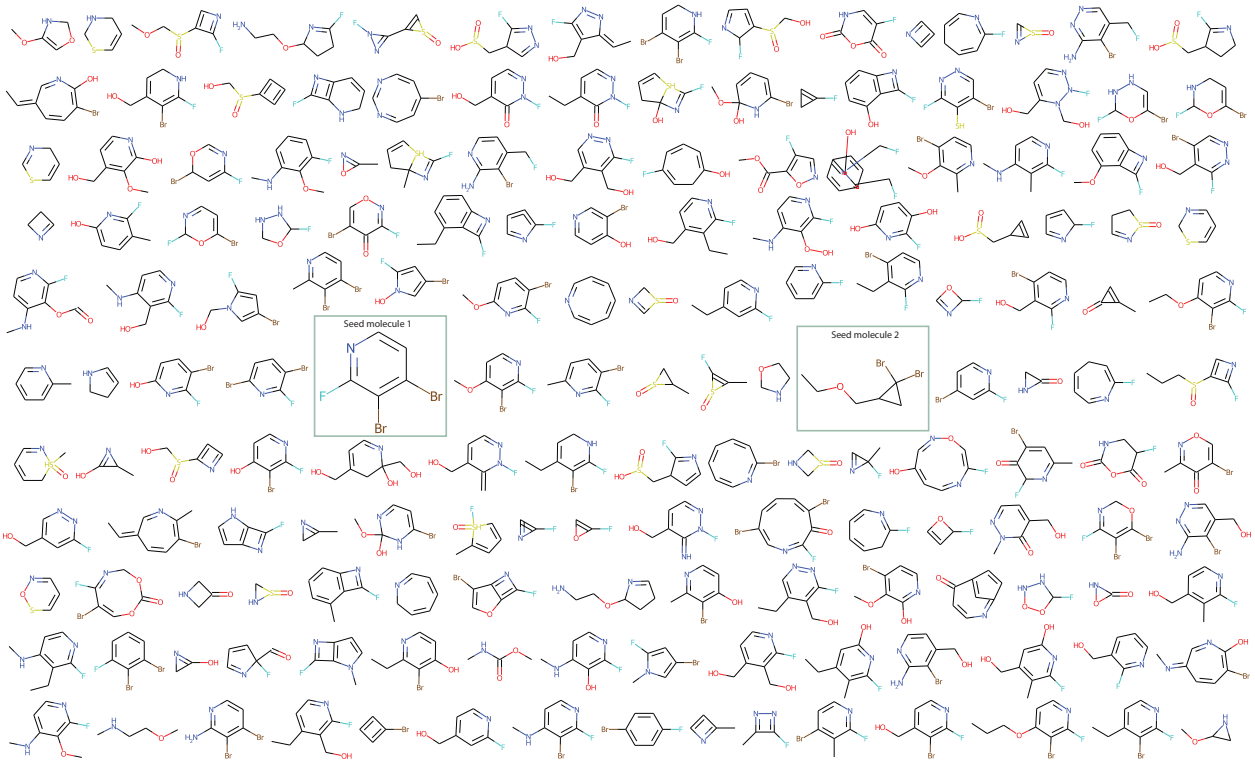


Figure 8: Example sampling of two molecules

Verification of encoder

Table 3: Exploring representation learning of transformer

No	Description	Test name	P Value	Reject H_0
1	Test of independence between molecular set 1 and molecular set 2	Mann-Whitney U Test	p=0.000	Yes
2	Test of independence between halfs of molecular set 1	Mann-Whitney U Test	p=0.432	No
3	Test of independence between halfs of molecular set 2	Mann-Whitney U Test	p=0.393	No

To verify the encoder we have set up the following experiment:

1. Selected 2 molecules

2. Generated a set of enumerated SMILES for each molecule

3. Encoded them into latent representations

4. Computed 2-norm on these vectors

5. Conducted statistical tests to verify the independence of distributions

Statistical tests have shown that latent representations of molecule 1 generated from enumerated SMILES have an independent distribution from the latent representations of molecule 2. See Table 3 for the details.

Adjusting weights

After verification of the encoder, the natural way of proceeding would be trying to generate intermediate representations of the molecules, and decode molecules that resemble both seed one and seed two. See Figure 5 for the illustration of intermediate representation in the context of latent space.

Conclusion

In summary, we have successfully applied a recent deep learning framework to the task of molecular generation using attention mechanisms. We have benchmarked the resulting Transmol method utilizing the MOSES benchmark. The results demonstrate a number of the advantages when using this attention-based methodology in comparison with earlier approaches.

References

(1) Reymond, J.-L. The Chemical Space Project. *Accounts of Chemical Research* **2015**, *48*, 722–730.

(2) Hartenfeller, M.; Schneider, G. Enabling future drug discovery by *de novo* design. *WIREs Computational Molecular Science* **2011**, *1*, 742–759.

(3) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. DOGS: Reaction-Driven de novo Design of Bioactive Compounds. *PLoS Computational Biology* **2012**, *8*, e1002380.

(4) Hartenfeller, M.; Eberle, M.; Meier, P.; Nieto-Oberhuber, C.; Altmann, K.-H.; Schneider, G.; Jacoby, E.; Renner, S. A Collection of Robust Organic Synthesis Reactions for *In Silico* Molecule Design. *Journal of Chemical Information and Modeling* **2011**, *51*, 3093–3098.

(5) Segler, M. H. S.; Waller, M. P. Modelling Chemical Reasoning to Predict and Invent Reactions. *Chemistry - A European Journal* **2017**, *23*, 6118–6128.

(6) Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* **2015**, *6*, 2326–2331.

(7) Jin, W.; Coley, C. W.; Barzilay, R.; Jaakkola, T. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA, 2017; p 2604–2613.

(8) Vanhaelen, Q.; Lin, Y.-C.; Zhavoronkov, A. The Advent of Generative Chemistry. *ACS Medicinal Chemistry Letters* **2020**, *11*, 1496–1505.

(9) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse Molecular Design Using Machine Learning: Generative Models for Matter Engineering. *Science* **2018**, *361*, 360.

(10) 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.* **2019**, *4*, 828.

(11) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-

Guzik, A. arXiv. 2018, preprint, arXiv:1705.10843v3; https://arxiv.org/abs/1705.10843v3.

(12) Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; Aspuru-Guzik, A. ChemRxiv. 2017, preprint, DOI:chemrxiv:10.26434/chemrxiv.5309668.v3.

(13) Harel, S.; Radinsky, K. Accelerating Prototype-Based Drug Discovery Using Conditional Diversity Networks. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18. London, United Kingdom, 2018; p 331.

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

(15) Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. Grammar Variational Autoencoder. Proceedings of the 34th International Conference on Machine Learning-Volume 70. 2017; p 1945.

(16) Lim, J.; Ryu, S.; Kim, J. W.; Kim, W. Y. Molecular Generative Model Based on Conditional Variational Autoencoder for de Novo Molecular Design. *J. Cheminf.* **2018**, *10*, 31.

(17) 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.* **2018**, *4*, 120.

(18) Jin, W.; Barzilay, R.; Jaakkola, T. Junction Tree Variational Autoencoder for Molecular Graph Generation. Proceedings of the 35th International Conference on Machine Learning. Stockholmsmässan, Stockholm Sweden, 2018; p 2323.

(19) Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A. A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference On Computer Vision (ICCV). 2017; p 2242.

(20) Maziarka, Ł.; Pocha, A.; Kaczmarczyk, J.; Rataj, K.; Warchoł, M. Mol-cycleGAN - a Generative Model for Molecular Optimization. Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions. Cham, 2019; p 810.

(21) Kang, S.; Cho, K. Conditional Molecular Design with Deep Generative Models. *J. Chem. Inf. Model.* **2019**, *59*, 43.

(22) Polykovskiy, D. et al. arXiv. 2020, preprint, arXiv:1811.12823v5; `https://arxiv.org/abs/1811.12823v5`.

(23) Zhumagambetov, R.; Kazbek, D.; Shakipov, M.; Maksut, D.; Peshkov, V. A.; Fazli, S. cheML.io: an online database of ML-generated molecules. *RSC Adv.* **2020**, *10*, 45189–45198.

(24) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling* **1988**, *28*, 31–36.

(25) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. Randomized SMILES strings improve the quality of molecular generative models. *Journal of Cheminformatics* **2019**, *11*, 71.

(26) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, u.; Polosukhin, I. Attention is All You Need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA, 2017; p 6000–6010.