LigGPT: Molecular Generation Using a Transformer-Decoder Model

Viraj Bagal,[1,2] **Rishal Aggarwal,** [2] **P. K. Vinod,** [2] **U. Deva Priyakumar** [2]

[1] Indian Institute of Science Education and Research, Pune 411 008, India
[2] International Institute of Information Technology, Hyderabad 500 032, India
viraj.bagal@students.iiserpune.ac.in, rishal.aggarwal@research.iiit.ac.in, vinod.pk@iiit.ac.in, deva@iiit.ac.in

Abstract

Application of deep learning techniques for molecular generation has been gaining traction for drug design. The representation of molecules in SMILES notation as a string of characters enables the usage of state of the art models in Natural Language Processing, such as the Transformers, for molecular design. Inspired from Generative Pre-Training (GPT) models, in this study, we train a Transformer-Decoder on the next token prediction task using masked self-attention for the generation of molecules. We show that our model has the best performance on the GuacaMol dataset and comparable performance on the MOSES dataset in generating valid, unique and novel molecules when benchmarked against other modern methods for molecular generation. Furthermore, we demonstrate that the model can be trained conditionally to control multiple properties of the generated molecules. As a potential real world application, the model can be used to generate molecules with desired properties thus catalysing the drug discovery process.

Introduction and Related Work

It has been postulated that in the whole chemical space, the total number of potential drug like candidates can range from 10^{23} to 10^{60} molecules(Polishchuk, Madzhidov, and Varnek 2013). However, only about 10^{8} molecules have been synthesized at least once(Kim et al. 2016). This disparity between synthesized and potential molecules beckons for the use of generative models that can model the distribution of molecules for efficient sampling.

Deep generative models have made great strides in modeling data distributions in general data domains such as Computer Vision(Goodfellow et al. 2014) and Natural Language Processing (NLP)(Vaswani et al. 2017). Therefore, such methods have also been adopted to model molecular distributions(Chen et al. 2018).

The earliest deep learning architectures for molecular generation involved the usage of Recurrent Neural Networks (RNNs) on molecular SMILES(Segler et al. 2018; Gupta et al. 2018). Such models have also previously been trained on large corpus of molecules and then focused through the usage of reinforcement learning(Popova, Isayev, and Tropsha 2018; Olivecrona et al. 2017) or transfer learning(Segler et al. 2018) to generate molecules of desirable properties and activity. Auto-encoder variants such as the Variational Auto-Encoder(VAE)(Liu et al. 2018; Kusner, Paige, and Hernández-Lobato 2017; Simonovsky and Komodakis 2018; Jin, Barzilay, and Jaakkola 2018; Lim et al. 2018; Pathak et al. 2020) and Adversarial Auto-Encoder(AAE)(Kadurin et al. 2017; Putin et al. 2018b; Polykovskiy et al. 2018; Hong et al. 2019) have also been employed for molecular generation. These models contain an encoder that encodes molecules to a latent vector representation and a decoder that maps latent vectors back to molecules. Junction Tree VAE (JT-VAE)(Jin, Barzilay, and Jaakkola 2018) is also a VAE model that represents molecules as graph tree structures. JT-VAE also ensures 100% validity of generated molecules by maintaining a vocabulary of molecular components that can be added at each junction of the molecule tree.

Generative Adversarial Networks (GANs) have also been successfully used for molecular design(Prykhodko et al. 2019; Guimaraes et al. 2017; Sanchez-Lengeling et al. 2017; De Cao and Kipf 2018; Putin et al. 2018a). ORGAN(Guimaraes et al. 2017) was the first usage of GANs for molecular generation. RANC(Putin et al. 2018a) introduced reinforcement learning alongside a GAN loss to generate molecules of desirable properties.

A recent development in NLP has been the Transformer model(Vaswani et al. 2017). Transformers use self and masked attention to efficiently gain context from all previous input and output tokens for its predictions. Thus, it has shown the state of the art performance in language translation tasks. Transformers consist of both encoder and decoder modules. The encoder module gains context from all the input tokens through self attention mechanisms. The decoder module gains context from both the encoder as well as previously generated tokens by attention. Using this context the decoder is able to predict the next token.

The decoder module has also been previously used independently for language modeling task and is known as the Generative Pre-Training model (GPT)(Radford et al. 2018; Brown et al. 2020; Radford et al.). In this work we train a smaller version of the GPT model to predict the next token for molecular generation. We call this model LigGPT. We demonstrate that the model can be trained conditionally to control multiple properties. Therefore, we propose that LigGPT is capable of generating molecules whose physiochemical properties are tuned to desired values while also

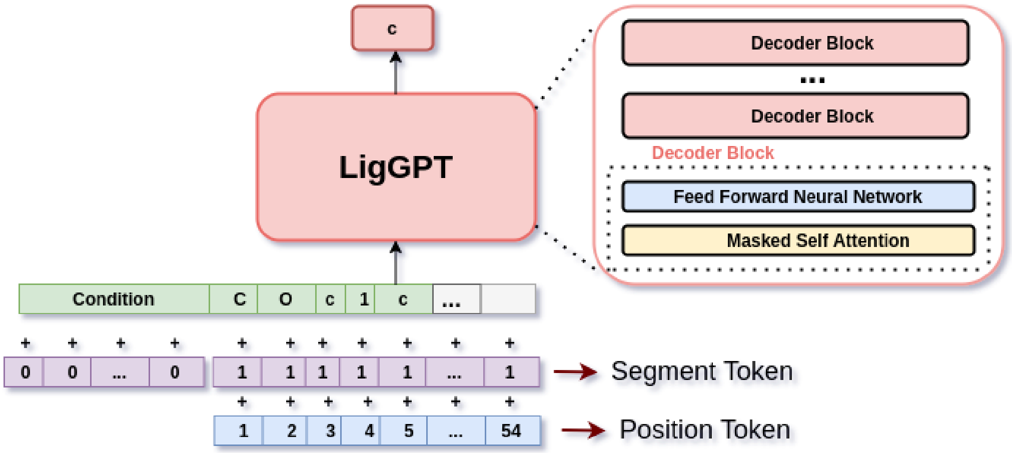


Figure 1: The network is trained on next token prediction task. For conditional training, the appropriate condition is transformed using linear layer and concatenated to the embedding of the SMILES representation of the molecule. Segment tokens allow the model to distinguish between the 'Condition' and the 'SMILES sequence'. Position tokens provide the information of the position of each SMILES token in the sequence. The embeddings of the SMILES token, segment token and position token are added and passed as input to the model.

maintatining high uniqueness and novelty.

Method

Our model is illustrated in Figure 1. The model is essentially a mini version of the Generative Pre-Training (GPT) model. Unlike GPT1 that has around 110M parameters, LigGPT has only around 6M parameters. This reduction in parameters reflects in low training time and thus a more efficient model. LigGPT comprises stacked decoder blocks, each of which is composed of a masked self-attention layer and feed forward layer. LigGPT consists of 8 such decoder blocks. Masked self-attention masks tokens that occur after the current time step. This is essential as during generation, the network would have access only to the tokens predicted in the previous time-steps. Moreover, instead of performing a single masked self-attention operation, each masked self-attention block performs multiple masked self-attention operations (multi-head attention) in parallel and concatenates the output.

We train this model on molecules represented as SMILES string. For this, we use a SMILES tokenizer to break up the string into a sequence of relevant tokens. Further, to keep track of the position of each token in the sequence, position tokens are assigned to each position. During conditional training, segment tokens are provided to distinguish between the condition and the molecule representation. All the tokens are mapped to the same space using respective embedding layers. All the embeddings are added and passed as input to the model.

Property conditions are also sent through a linear layer that maps the condition to a vector of 256 dimensions. The resultant vector is then concatenated at the start of the sequence of the embeddings of the SMILES tokens. The model is trained such that the predicted tokens are a result of attention to both the previous tokens as well as the conditions. During generation a start token of a single carbon atom is provided to the network along with the conditions.

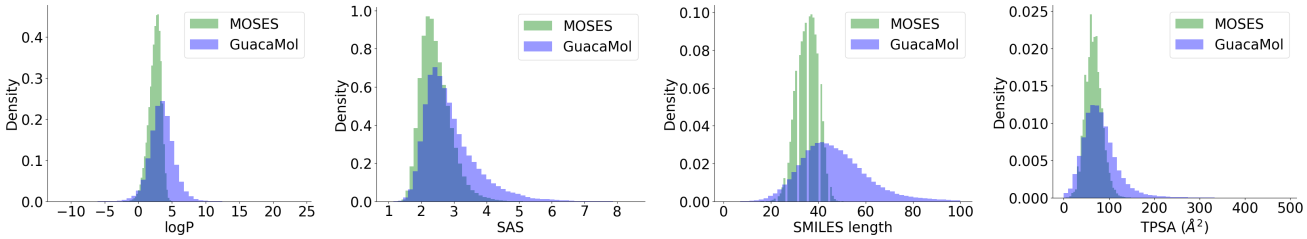
Datasets

In this work, we use two benchmark datasets, MOSES and GuacaMol for training and evaluation of our model. MOSES is a dataset composed of 1.9 million clean lead-like molecules with molecular weight ranging from 250 to 350 Daltons, number of rotatable bonds lower than 7 and XlogP below 3.5. GuacaMol on the other hand is a subset of the ChEMBL 24(Gaulton et al. 2017) and contains 1.6 million molecules. To calculate molecular properties, we use the RDKiT toolkit(Landrum 2013). As shown in Figure2, molecular property distributions in the GuacaMol dataset have a higher range making it suitable to test property conditional generation. Following are the properties considered in the study:

- **logP**: the logarithm of the partition coefficient. Partition coefficient compares the solubilities of the solute in two immiscible solvents at equilibrium. This helps in assessing the bioavailability of the drug molecule
- **Synthetic Accessibility score (SAS)**: measures the difficulty of synthesizing a compound. It is a score between 1 (easy to make) and 10 (very difficult to make).
- **Topological Polar Surface Area (TPSA)**: the surface sum over all polar atoms. It measures the drug's ability to permeate cell membranes. Molecules with TPSA greater than 140 $Å^2$ tend to be poor at permeating cell membranes.

Results

Before explaining the experimental results, we describe the metrics used to evaluate the models:



(a) (b) (c) (d)

Figure 2: Distribution of properties in MOSES and GuacaMol datasets.

- **Validity**: the fraction of generated molecules that are valid. We use RDKit for validity check of molecules. Validity measures how well the model has learnt the SMILES grammar and the valency of atoms.
- **Uniqueness**: the fraction of valid generated molecules that are unique. Low uniqueness highlights mode collapse.
- **Novelty**: the fraction of valid unique generated molecules that are not in the training set. Low novelty is a sign of overfitting.
- **Internal Diversity (IntDiv$_\mathbf{p}$)**: measures the diversity of the generated molecules using Tanimoto similarity (T) of each pair of molecules in the generated set (S).

$$IntDiv_p(S) = 1 - \sqrt[p]{\frac{1}{|S|^2} \sum_{s1,s2 \in S} T(s1,s2)^p}$$

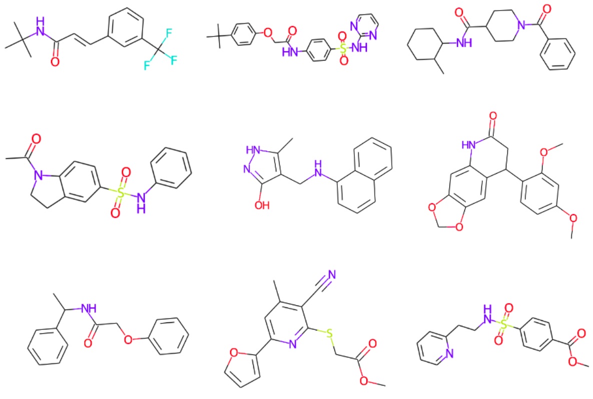


Figure 3: Generated molecules.

Unconditional Generation

We compare the performance of LigGPT on the MOSES dataset to that of CharRNN, VAE, AAE, LatentGAN(Prykhodko et al. 2019) and JT-VAE. JT-VAE uses graphs as input while the others use SMILES. To get the optimal model for each dataset we check the generative performance for several sampling temperature values between 0.7 and 1.6. We notice that the model performs best at a temperature of 1.6 for MOSES and 0.9 for GuacaMol. We report the optimal model performance on each dataset in Table 1 and Table 2.

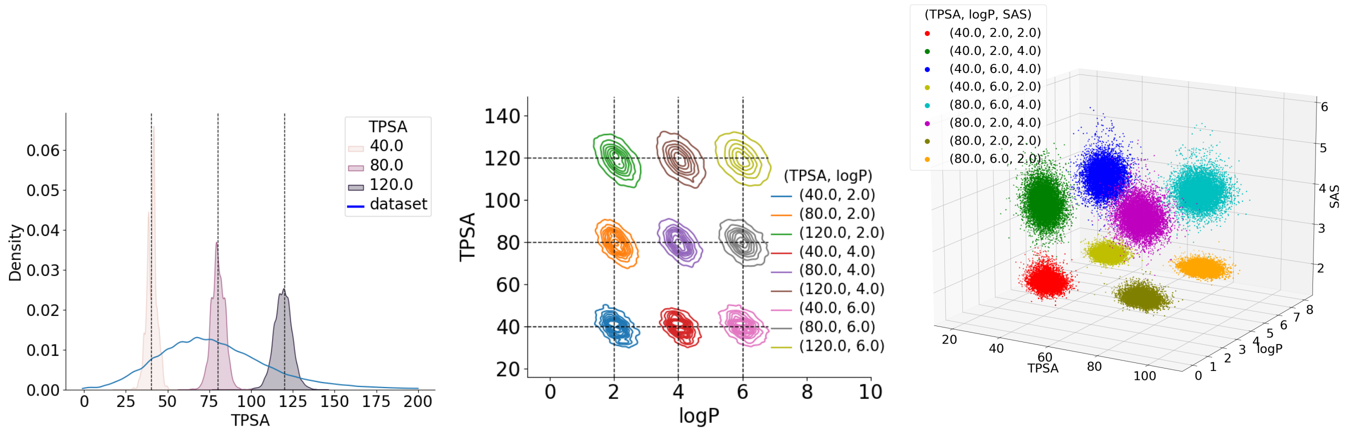
Models	Validity	Unique@10K	Novelty	IntDiv$_1$	IntDiv$_2$
CharRNN	0.975	**0.999**	0.842	0.856	0.85
VAE	0.977	0.998	0.695	0.856	0.85
AAE	0.937	0.997	0.793	0.856	0.85
LatentGAN	0.897	0.997	**0.949**	0.857	0.85
JT-VAE	**1.0**	**0.999**	0.9143	0.855	0.849
LigGPT	0.9	**0.999**	0.941	**0.871**	**0.865**

Table 1: Unconditional training on MOSES dataset. Temperature 1.6 was used.

Models	Validity	Unique	Novelty
SMILES LSTM	0.959	**1.0**	0.912
AAE	0.822	**1.0**	0.998
ORGAN	0.379	0.841	0.687
VAE	0.870	0.999	0.974
LigGPT	**0.986**	0.998	**1.0**

Table 2: Unconditional training on GuacaMol dataset. Temperature 0.9 was used.

On the MOSES benchmark, LigGPT performs the best in terms of the two internal diversity metrics. This indicates that even though LigGPT learns from the same chemical space as other models, it is better than others at generating molecules with lower redundancy. In case of validity, JT-VAE always generates a valid molecule because it checks validity at every step of generation. Barring JT-VAE, we observe that CharRNN, VAE and AAE have high validity but low novelty. Compared to these three & JT-VAE, LigGPT has lower validity but much higher novelty. We find that the performance of LigGPT is comparable to LatentGAN. LatentGAN involves training of an autoencoder followed by the training of GAN on the latent space of the trained autoencoder. This is a 2-step process while on the other hand, LigGPT is trained end-to-end. We observe that LigGPT's validity, uniqueness and novelty is similar to LatentGAN's. On the GuacaMol benchmark, we see that LigGPT is easily the most preferred method when compared to other methods(Guimaraes et al. 2017; Brown et al. 2019) tested on it. It returns very high validity, uniqueness and novelty scores on generation with a temperature of 0.9. We believe this boost



(a)

(b)

(c)

Figure 4: Distribution of properties of generated molecules conditioned on the particular properties. The unit of TPSA is $\mathring{A}^2$.

Condition	Validity	Uniqueness	Novelty	MAD_TPSA	MAD_logP	MAD_SAS
TPSA	0.992	0.966	1.0	3.339	-	-
TPSA+logP	0.989	0.942	1.0	3.528	0.227	-
TPSA+logP+SAS	0.99	0.874	1.0	3.629	0.254	0.158

Table 3: Multi-property conditional training on GuacaMol dataset. Temperature 0.9 was used.

in performance, as compared to MOSES, is due to a larger diversity in molecules in the GuacaMol dataset. Moreover, even though GuacaMol dataset has larger molecules as compared to MOSES dataset, LigGPT generates molecules with very high validity. This indicates that LigGPT handles long range dependencies very well.

Property Conditional Generation

Since GuacaMol has a larger range in property values, we test the model's ability to control molecular properties on it. While we use only logP, SAS(Ertl and Schuffenhauer 2009) and TPSA for property control, the model can be trained to learn any molecular property. For each condition we generate 10,000 molecules to evaluate property control.

Distribution of TPSA from the generated molecules for single property control are visualized in Figure 4(a). As seen in the figure, generated values deviate only slightly from the intended values. Next, for multi-property control we check the model's capacity to control two (Figure 4(b)) and three properties (Figure 4(c)) at a time. We see well separated clusters centered at the desired property value indicating high accuracy. The average values of validity, uniqueness, novelty and Mean Absolute Difference (MAD) scores for each condition are reported in Table 3. The model's accuracy is further exemplified by the low MAD scores despite having to control multiple properties at a time.

Conclusion

In this work, we designed a Transformer-Decoder model called LigGPT for molecular generation. This model utilises masked self-attention mechanism that make it simpler to learn long range dependencies between string tokens. This is especially useful to generate valid SMILES strings that satisfy chemical valencies. We see through our benchmarking experiments that LigGPT shows very high validity, uniqueness and novelty scores compared to the state of the art methods. LigGPT is able to show good performance on the MOSES and GuacaMol datasets with it outperforming all other methods benchmarked on the GuacaMol dataset. We also show that the model learns higher level chemical representations through molecular property control. LigGPT is able to generate molecules with property values that deviate only slightly from the exact values that are passed by the user for both single and multi-property control. Consequently, we believe that the LigGPT model may evolve to be a strong architecture to be used by itself or incorporated into other molecular generation techniques.

References

Brown, N.; Fiscato, M.; Segler, M. H.; and Vaucher, A. C. 2019. GuacaMol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling* 59(3): 1096–1108.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* .

Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; and Blaschke, T. 2018. The rise of deep learning in drug discovery. *Drug discovery today* 23(6): 1241–1250.

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

Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; et al. 2017. The ChEMBL database in 2017. *Nucleic acids research* 45(D1): D945–D954.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27: 2672–2680.

Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; and Aspuru-Guzik, A. 2017. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv preprint arXiv:1705.10843* .

Gupta, A.; Müller, A. T.; Huisman, B. J.; Fuchs, J. A.; Schneider, P.; and Schneider, G. 2018. Generative recurrent networks for de novo drug design. *Molecular informatics* 37(1-2): 1700111.

Hong, S. H.; Ryu, S.; Lim, J.; and Kim, W. Y. 2019. Molecular Generative Model Based on an Adversarially Regularized Autoencoder. *Journal of Chemical Information and Modeling* 60(1): 29–36.

Jin, W.; Barzilay, R.; and Jaakkola, T. 2018. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364* .

Kadurin, A.; Nikolenko, S.; Khrabrov, K.; Aliper, A.; and Zhavoronkov, A. 2017. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular pharmaceutics* 14(9): 3098–3104.

Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A.; et al. 2016. PubChem substance and compound databases. *Nucleic acids research* 44(D1): D1202–D1213.

Kusner, M. J.; Paige, B.; and Hernández-Lobato, J. M. 2017. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925* .

Landrum, G. 2013. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling.

Lim, J.; Ryu, S.; Kim, J. W.; and Kim, W. Y. 2018. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics* 10(1): 1–9.

Liu, Q.; Allamanis, M.; Brockschmidt, M.; and Gaunt, A. 2018. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems* 31: 7795–7804.

Olivecrona, M.; Blaschke, T.; Engkvist, O.; and Chen, H. 2017. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics* 9(1): 48.

Pathak, Y.; Juneja, K. S.; Varma, G.; Ehara, M.; and Priyakumar, U. D. 2020. Deep learning enabled inorganic material generator. *Physical Chemistry Chemical Physics* 22(46): 26935–26943.

Polishchuk, P. G.; Madzhidov, T. I.; and Varnek, A. 2013. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of computer-aided molecular design* 27(8): 675–679.

Polykovskiy, D.; Zhebrak, A.; Vetrov, D.; Ivanenkov, Y.; Aladinskiy, V.; Mamoshina, P.; Bozdaganyan, M.; Aliper, A.; Zhavoronkov, A.; and Kadurin, A. 2018. Entangled conditional adversarial autoencoder for de novo drug discovery. *Molecular pharmaceutics* 15(10): 4398–4405.

Popova, M.; Isayev, O.; and Tropsha, A. 2018. Deep reinforcement learning for de novo drug design. *Science advances* 4(7): eaap7885.

Prykhodko, O.; Johansson, S. V.; Kotsias, P.-C.; Arús-Pous, J.; Bjerrum, E. J.; Engkvist, O.; and Chen, H. 2019. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics* 11(1): 74.

Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; and Zhavoronkov, A. 2018a. Reinforced adversarial neural computer for de novo molecular design. *Journal of chemical information and modeling* 58(6): 1194–1204.

Putin, E.; Asadulaev, A.; Vanhaelen, Q.; Ivanenkov, Y.; Aladinskaya, A. V.; Aliper, A.; and Zhavoronkov, A. 2018b. Adversarial threshold neural computer for molecular de novo design. *Molecular pharmaceutics* 15(10): 4386–4397.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. ???? Language models are unsupervised multitask learners .

Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; and Aspuru-Guzik, A. 2017. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC) .

Segler, M. H.; Kogej, T.; Tyrchan, C.; and Waller, M. P. 2018. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science* 4(1): 120–131.

Simonovsky, M.; and Komodakis, N. 2018. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, 412–422. Springer.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.