# Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC)

Benjamin Sanchez-Lengeling,[†,¶] Carlos Outeiral,[†,¶] Gabriel L. Guimaraes,[†] and Alán Aspuru-Guzik[*,†,‡]

†*Department of Chemistry and Chemical Biology, Harvard University, 12 Oxford Street, Cambridge, Massachusetts 02138, United States*

‡*Canadian Institute for Advanced Research (CIFAR) Senior Fellow*

¶*Contributed equally to this work*

E-mail: aspuru@chemistry.harvard.edu

**Abstract**

Molecular discovery seeks to generate chemical species tailored to very specific needs. In this paper, we present *ORGANIC*, a framework based on *Objective-Reinforced Generative Adversarial Networks* (ORGAN), capable of producing a distribution over molecular space that matches with a certain set of desirable metrics. This methodology combines two successful techniques from the *machine learning* community: a Generative Adversarial Network (GAN), to create non-repetitive sensible molecular species, and Reinforcement Learning (RL), to bias this generative distribution towards certain attributes. We explore several applications, from optimization of random physicochemical properties to candidates for drug discovery and organic photovoltaic material design.

# Introduction

In the past century, the impact that molecules such as penicillin, progestin or azidothymidine have had in human society is only comparable to the monumental effort (in terms of either time, money or luck) that their finding demanded. Even nowadays, a new material requires an estimated time of 20 years before it is ready to be commercialized. This fact has inspired *molecular discovery*, a scientific current that seeks to bypass the need of serendipity using a plethora of techniques. For molecular discovery, we need to navigate chemical space, the set of all possible compounds. Navigating such a space is hard, the space is incredibly large and discrete, small changes in a molecule can change radically its properties [1]. This challenge has prompted the emergence of large-scale joint projects, such as the Materials Genomic Project in the United States, or the NOMAD laboratory in Europe.

Current strategies to explore and discover molecules lie with *high-throughput virtual screening* [2] or discrete local search methods such as genetic algorithms [3–5]. With high-throughput screening, an enormous library of compounds is generated, and then successively filtered using increasingly expensive computational techniques to determine if a given species has the desired properties. This strategy has achieved many important triumphs in pharmaceutical contexts [6, 7], design of organic photovoltaics [8, 9], OLEDs [10] and flow batteries [11, 12], among many others. However, this approach has many drawbacks, mainly that its coverage of molecular space is limited to the combinations of initial fragment libraries.

Materials discovery can be posed as an inverse-design problem, where a set of properties is specified and then fit candidates are generated. Nowadays, feasibility of inverse design has been accelerated partly due to new developments from the Artificial Intelligence community: Generative Adversarial Networks [13, 14], text sequence generation models [15–18] and Reinforcement Learning (RL) [19–21].

Their application onto chemical space has brought new ways to explore, represent and calculate molecules. Recently Bombarelli et al. employed a variational autoencoder for molecular generation[22] which allows to explore and optimize in a continuous latent space,

while also being able to sample molecules from such space. Kadurin et al. reported an adversarial autoencoder for drug design[23] and Gilmer et al. has demonstrated Message Passing Neural Networks for fast, accurate prediction of properties [24].

In this work we present *ORGANIC*, an implementation of Objective-Reinforced Generative Adversarial Networks for Inverse-design Chemistry based on the ORGAN algorithm [25]. As we will show, this code[1] is able to optimize a distribution over molecular space according to certain desired properties.

# Methods

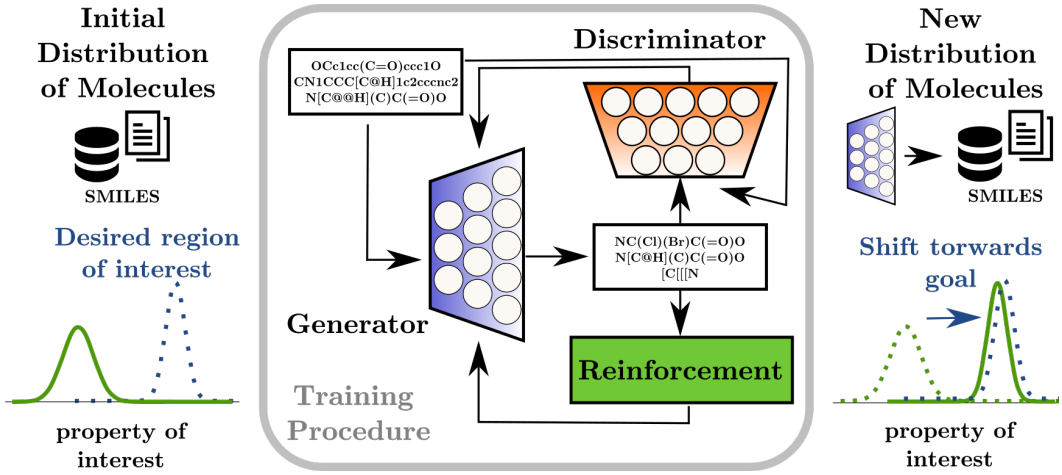


Figure 1: Usage of ORGANIC illustrated. In the training procedure we show the three fundamental components: a generator, a discriminator, and a reinforcement metric. Arrows indicate the flow of inputs and outputs between networks.

In Figure 1, the general scheme of *ORGANIC* is illustrated. It is a chemistry-orientated implementation of the Objective-Reinforced Generative Adversarial Networks paradigm which combines a Generative Adversarial Network with a Reinforcement Learning component.

The model is composed of three key elements: a Generator $G$, a Discriminator $D$ and a Reinforcement component $R$. On one side, the discriminative network determines whether a molecule is likely to come from the initial distribution (positive) or not (negative). The

[1]The *ORGANIC* code is available online at `https://github.com/aspuru-guzik-group/ORGANIC`

reinforcement provides a quality metric $R(x) \in [0, 1]$ that will quantify the desirability of a given molecule $x$, where 1 is meant to represent the desired shift in properties and 0 an undesired change. Finally, the generator has the task of generating molecules that maximizes the objective function which is a linear combination of the discriminator component and the reinforcement component, parametrized by a tunable parameter $\lambda$.

In this way, the adversarial approach is meant to keep the generation of molecules similar to the initial distribution of data, while the reinforcement learning biases this generation towards the maximization of the reward (and, if the metric is well-defined, some desirable properties).

## Molecular encoding

The molecules were represented as characters strings using the SMILES encoding [26] and then hot-encoded as binary matrices. SMILES are able to capture the topology of a molecular species via defined grammar rules. In a string, the characters represent atoms, bond types, cycles and branches within a molecular graph. In this manner, the set of characters and rules already contain some heuristics about how molecules are built. However, this encoding also implies several inconveniences, such as the existence of invalid representations. SMILES strings can be valid 'N#Cc1cc[nH]n1" or invalid like "[C[[[N", meaning they can represent a molecule under the grammar defined by SMILES. Furthermore, the representation is non-unique, several strings can map to the same molecule (although there exists canonization algorithms [27]).

For the hot-encoding, we build a dictionary of $m$ possible characters, each is assigned an index for the hot-encoding. For example, the $i-th$ character in a string $x_i$, will get encoded to a binary vector of length $m$, with 0's except in the index that maps to the character represented. A maximum length $T$ is decided based on the training set and on the expected size of the string. In this way, each molecule is encoded to a $n \times T$ binary matrix, which can be converted back to a SMILES and then to a molecule.

In our work we employ the RDKit [28] package for manipulation and verification of SMILES.

## Adversarial approach

Although the generation of valid SMILES can be trivial if one relies on simple permutations of carbons and oxygen atoms, such a strategy can lead to millions of different possible molecules which might not be of any interest for a given problem domain. In order to generate molecular species that resemble a given initial distribution, a key strategy is to make use of adversarial training.

Generative Adversarial Networks (GANs)[13] are a generative model that aims to minimize the divergence between a real data distribution $p_{data}$ and the distribution $p_{synth}$ generated by an implicit generative model $G$. The main idea is that two different neural networks play a game against each other: given an initial training distribution $p_{data}$, the generator $G$ samples $x$ from a distribution $p_{synth}$ generated with random noise $z$, while a discriminator $D$ looks at samples, either from $p_{syntetic}$ or from $p_{data}$, and tries to classify their identity ($y$) as either real ($x \in p_{data}$) or fake ($x \notin p_{data}$).

The GAN setting is then formulated as a zero sum game where the value is the cross-entropy loss between the discriminator's prediction and the true identity of the samples. This is implemented in practice as a minmax optimization problem, one model is optimized with respect to the performance of another model alternatively:

$$\min_G \max_D \mathbf{E}_{x\sim p_{data}(x)} \left[logD(x)\right] + \mathbf{E}_{z\sim p_{synthetic}(z)} \left[log(1 - D(G(z)))\right]$$

In our setting the discriminator $D$ is a Convolutional Neural Network (CNN)[29] parameterized= by $\phi$, while the generator $G$ is an RNN parameterized by $\theta$ using LSTM (Long Short Term Memory) cells [30] that generates sequences $X_{1:T} = (x_1, ..., x_T)$ of length $T$, which in our case might represent valid SMILES strings.

With the correct training procedures, GANs are able to work remarkably well for continuous variable outputs since the gradient is propagated between networks during optimization. For discrete variable outputs, this proves difficult since these objects are sampled from a multinomial distribution, with probabilities given by the output of a softmax function, which is not differentiable and hence cannot be optimized with respect to each other easily. [18]

The generator (which has previously been trained on the training set using MLE) generates batches of molecules, which are analyzed by the discriminator and the metric; then the former is trained to simultaneously fool the discriminator and maximize the reward.

## Reinforcement Learning

It is only with recent developments that GANs have been adapted for discrete objects. One of the strategies is to bypass the generator differentiation problem by treating the generation of discrete sequences as a stochastic policy in an RL setting. The gradient is in this case approximated as a gradient policy update [31].

With a policy gradient, we treat $G$ as an agent in an RL game where we consider states $s$, actions $a$ from an action space $A$ and a reward function $Q$. A state $s_t$ is an already generated partial sequence of characters $X_{1:t}$. We have an action space $A$ that encompasses all possible characters to select for the next character $x_{t+1}$. Next a reward function $Q(s, a)$ that represents the expected reward for taking action $a$ in state $s$. Each episode is the completion of a fully generated sequence of fixed length $T$, which is rewarded with he function $R_T(X)$. The agent's stochastic policy is given by $G_\theta(x_t|X_{1:t-1})$ and we wish to maximize its expected long term reward $J(\theta)$:

$$J(\theta) = E[R_T|s_\theta, \theta] = \sum_{x_1 \in X} G_\theta(x_1|s_\theta) \cdot Q(s_\theta, x_1) \tag{1}$$

For any full sequence $X_{1:T}$, we have $Q(s = X_{1:T-1}, a = X_T) = R_T(X)$. In order to calculate which action $a$ is best for partial sequences $s_t$ at intermediate timesteps, we need

to consider the expected future reward when the sequence is completed. To calculate $Q$ in such cases, we perform $N$-time Monte Carlo search with the canonical rollout policy $G_\theta$. That means from a partial sequence $X_{1:t}$, we sample stochastically from 1 to $N$, completed sequences $X^n_{t+1:T}$ via the policy $G_\theta$.

This formulation allows us to choose the source of completion reward $R$, either from the discriminator or from a quality metric. In ORGAN, the parameter $\lambda$ controls the contribution of the source of each reward, where $\lambda = 0$ represents a complete RL approach and $\lambda$ represents a complete GAN training. Typically one will want to use a value in-between.

The quality metric for RL can be defined from the myriad of functions designed for properties of interest in molecules. Some examples include Log P [32], Synthetic Accessibility Score [33], Natural Product-likeliness [34], Chemical Beauty (QED)[35] and presence or absence of substructures that can be calculated with chemoinformatic tools [28]. Complex properties such as HOMO-LUMO bandgap energies, Photoelectric Conversion Efficiency (PCE) and Redox potentials can be calculated with quantum chemistry methods. In practice, these can be estimated order of magnitude much faster, but less accurately, with machine learning regression tools such as Neural Networks and Gaussian Processes.

# Experiments and discussion

This section reports the computational experiments performed with *ORGANIC*; it comprises three examples which simultaneously tackle interesting chemical problems and address different kinds of desired chemical distributions.

## Datasets

In our experiments we have made use of the following datasets:

- *Small molecules*, a 5,000 molecules subset of the dataset of roughly 134 thousand stable small molecules, which is itself a subset of all molecules with up to nine heavy atoms

(CONF) out of the GDB-17 universe of 166 billion organic molecules [36][37].

- *Drug-like compounds*, a subset of 15,000 drug-like molecules randomly selected from the ZINC database of 35 million commercially-available compounds[38][39].

- *Organic photovoltaics*, a subset of 15,000 non-fullerene acceptors proceeding from the Harvard Clean Energy Project [8]. It comprises the compounds with best Power Conversion Efficiencies.

## Melting point

As an initial proof-of-concept, we attempted to optimize an uncommon target property: the *melting point*. To this purpose, a feed-forward neural network with two 300-neuron hidden layers was trained on 15,000 examples randomly extracted from the valid data points of Jean-Claude Bradley open melting point dataset [40]; the molecules were encoded as 4096-bit radius 12 Morgan fingerprint vectors.

*ORGANIC* was trained for 120 epochs on the *small molecules* training set. In each training epoch, the melting points were computed using the former neural network, and then remapped to the [0, 1] interval for use as a metric; in this way, the model rewards the generated molecules with the highest melting points.

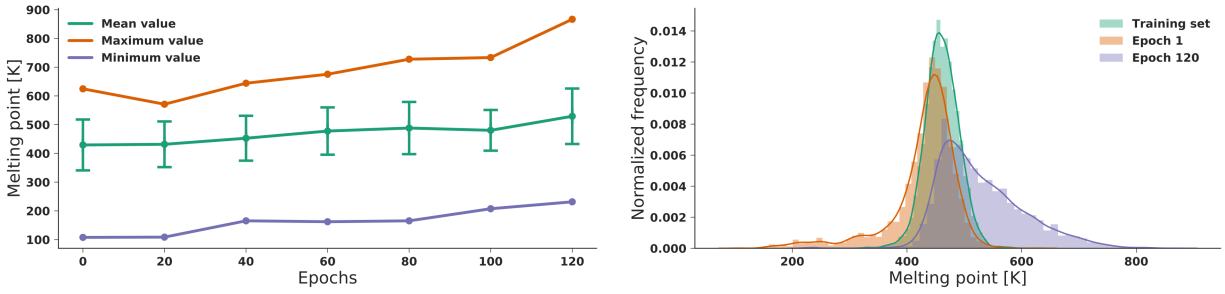


Figure 2: Right: evolution of the mean score during the 200 epochs of training for chemical beauty. Left: distribution of melting point corresponding to the first and last epochs, as compared to the training set.

The evolution of the training is shown in Figure 2. As it can be readily seen, the model

promptly learns to extend the molecular distribution towards greater melting points; although the mean is barely displaced (and this may have more to do with the actual structure of chemical space), both tails of the distribution increase dramatically, around 200 degrees more.

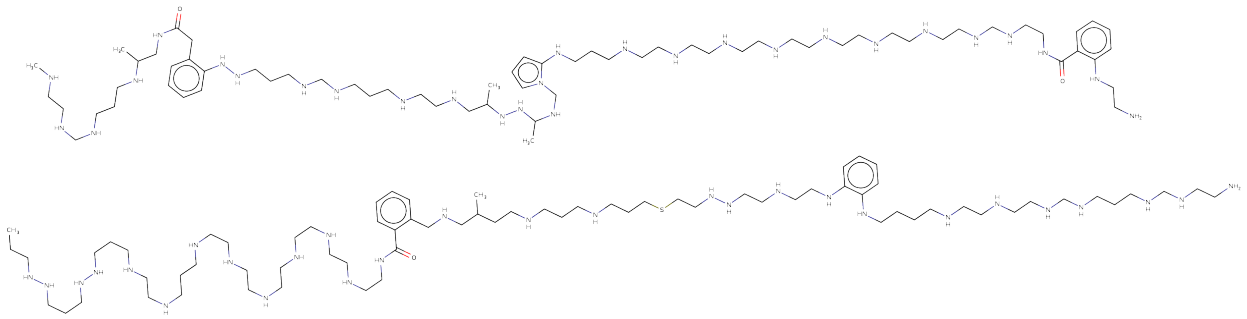

Figure 3: Two organic molecules with melting points above 800 K. Both combine an enormous size with the presence of hydrogen-bond donors and acceptors, which reflects the chemical patterns learned by ORGANIC. Most of the valid molecules generated after 120 epochs of training have similar structures.

What is more, *ORGANIC* discovers patterns in molecular structure that match with chemical intuition. As shown in Figure 3, the generated molecules combine many factors (long hydrocarbon chains, as well as hydrogen-bond donors and acceptors) that enhance non-covalent interactions between different molecules, or features such as rigid rings that hinder packing, that result in a higher melting point. It should be noted that in this case, we were able to generate molecules much bigger than those in the initial training set.

## Drug discovery

We tested the performance of ORGANIC in the drug dataset, using two drug-likeness indicators: the quantitative estimate of drug-likeness, here dubbed "chemical beauty" [35] and Lipinski's rule-of-five [41]. The first is already contained in the [0, 1] interval, but in the case of Lipinski's rule we designed a metric which assigned a 0.25 reward per followed rule. In both cases, a 200-epoch optimization was carried out.

The evolution of the training is shown in Figure 4 for chemical beauty.

Regarding chemical beauty, it is worth mentioning that the number of molecules with

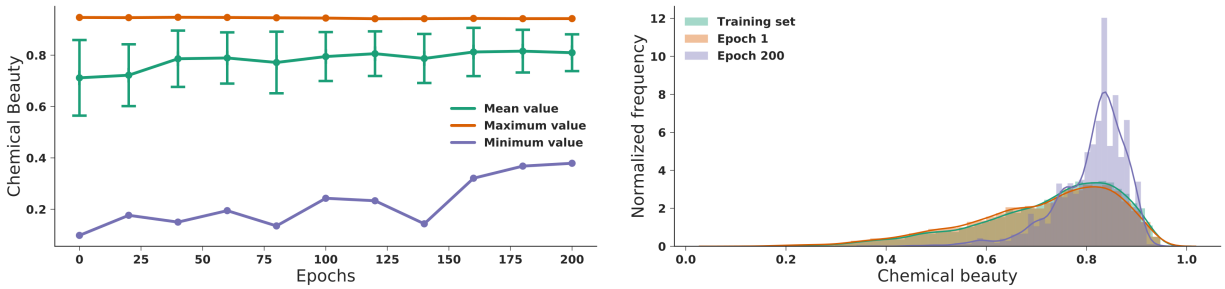


Figure 4: Right: evolution of the mean score during the 200 epochs of training for chemical beauty. Left: distribution of chemical beauty corresponding to the first and last epochs, as compared to the training set.

maximum rewards after 200 epochs is slightly lower than in the training set. This might have two sources. First, as the metric has a fixed maximum value (in contrast to the previous example, where the melting point has no practical limit, and the metric is updated as a remapping in every epoch) which is very difficult to find in chemical space, the model tends to maximize the 'integral of the reward' maximizing the number of molecules in the 0.8-0.9 range, much more abundant. On the other hand, as can be seen in Figure 4, this particular epoch is a bit noisy.

Regarding Lipinski's rule, ORGANIC is not able to improve the training set's distribution, which we have attributed to the categorical character of the metric.

Moreover, we compared the molecules generated during training with the list of FDA-approved drugs [42], and found 148 coincidences for the chemical beauty and 207 for the Lipinski's rule-of-five experiments; these include popular compounds, such as salicylic acid, benzocaine, paracetamol or sulfanilamide. Although these findings can be partially attributed to randomness, the vastness of the explored chemical space, and the fact that all of these drugs present high scores in the established metrics makes this fact at least worth mentioning.

## Non-fullerene acceptors for OPVs

Now we address another popular inverse-design problem in the recent literature: Non-fullerene electron acceptors for use in organic solar cells. To this matter, we trained a feed-forward

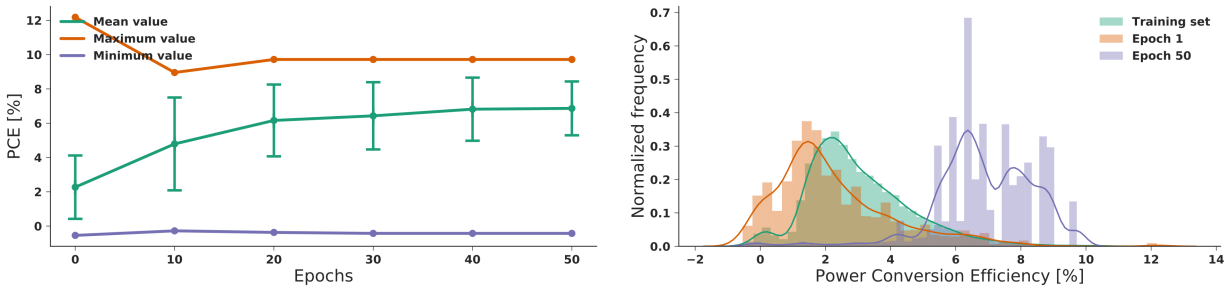


Figure 5: Right: evolution of PCE across the 50 epochs of training. Left: distribution of the PCE corresponding to the first and last epochs, as compared to the training set.

neural network with two 300-neuron hidden layers on 15,000 examples randomly extracted from a data set pertaining to the Harvard Clean Energy Project. Then we trained ORGANIC on the OPV dataset. In each training epoch, the PCE was computed and remapped to the [0, 1] interval The results are gathered in Figure 5. It is easy to infer that the average values of the PCE are improved really quickly, although the maximum varies greatly (for example, the maximum value, around 12%, is found in the first epoch). However, these results can be easily attributed to the nature of the metric: a neural network which has been trained on a combinatorial library and might not generalize easily to less structured molecules. Furthermore, the error in estimation is quite large, with a mean of 2 percentual points of PCE. More accurate estimators are likely to produce more certain results.

## Limitations of the framework

In the past sections we have shown several successful applications of our model. We will now address some of its limitations.

A major drawback of the ORGAN paradigm is the amount of non-valid molecules. In our experiments with *ORGANIC*, we have found the ratio of non-valid molecules to vary greatly in the range 0.2-99.9%. What is more, from the portion of valid molecules, it is possible to find a lot of repetitive patterns. Both cases depend greatly on the training set and the optimized metrics.

However, the main reason for this behavior is the landscape of the chemical space. As an

example, in the optimization of non-fullerene acceptors, we found percentages of non-validity in the range 7-99%. This is due to the *roughness* of the chemical space, where small changes in the molecular structure (for example, a disruption of the conjugation) can have dramatic effects on the structure.

Another possible point of failure lies when the density of molecules for a certain criteria is very sparse. For example, if we seek to generate symmetric molecules, the density of symmetric molecules with respect to non-symmetric molecules is very low and so it easy for the generator to collapse under training. In part because symmetry is not an apparent feature in the SMILES representation of molecules.

On the other hand, we have performed little variations on the $\lambda$ parameter described in the *Methods* section. After several tests, and checking previous results [25], we found 0.5 to be the most fitting value, but further work should be dedicated to this hyperparameter (and, for example, its variations across training). As a rule of thumb, any value that is not 1.0 or 0.0 will give reasonable results.

It should also be noted that SMILES grammar rules already encode heuristics about chemistry, and so part of the success of ORGANIC lies in the representation of molecules as strings. There are possibly better textual representations for different chemistry domains [43], future work will also look at overcoming SMILES validation problems with grammar aware networks[44].

# Conclusions and future work

In this paper, we have shown how a chemically oriented implementation of *Objective-Reinforced Generative Adversarial Networks* which can perform molecular generation biased towards some desired chemical properties. Our code is an open source, all figures can be easily reproduced from the repository and have been thoroughly designed so non-expert can run a generation task with as little as 10 lines of python commands. what is more, the computational requires

are modest and can be run on a common personal computer.

Through several examples, addressing many of the current topics of molecular discovery, we have shown the performance of ORGANIC as compared with a genetic algorithm. All the results reveal that we are able to effectively shift the molecular distribution towards user-specified properties.

We hope that this implementation will grow to become a standard in molecular generation. In the recent future, we seek to extend this work to combine the generative process with high-throughput screening pipelines that could act as more reliable reward functions. There is also the possibility of augmenting this approach with the recent developments of attention networks and latent space exploration with text sequence models [45, 46].

*ORGANIC* has been carefully designed to be used by non-expert users, and can be accessed at `https://github.com/aspuru-guzik-group/ORGANIC`.

## Acknowledgement

We thank Dr. Anders Fröseth for his generous support for this work. The authors thank Harvard Research Computing for their support on using the Odyssey cluster, as well as Professor Steven Lopez for providing the OPV dataset, and Professor Rafael Gómez-Bombarelli for letting them use his genetic algorithm for benchmarking. Carlos Outeiral also thanks Fundación María Cristina Masaveu Peterson for supporting financially his internship at Harvard.

## References

(1) Kirkpatrick, P.; Ellis, C. Chemical space. *Nature* **2004**, *432*, 823–823, DOI: 10.1038/432823a.

(2) Pyzer-Knapp, E. O.; Suh, C.; Omez-Bombarelli, R.; Aguilera-Iparraguirre, J.; Aspuru-

Guzik, A. What Is High-Throughput Virtual Screening? A Perspective from Organic Materials Discovery Keynote Topic. *Annu. Rev. Mater. Res* **2015**, *45*, 195–216, DOI: 10.1146/annurev-matsci-070214-020823.

(3) Balamurugan, D.; Yang, W.; Beratan, D. N. Exploring chemical space with discrete, gradient, and hybrid optimization methods. *J. Chem. Phys.* **2008**, *129*, 174105, DOI: 10.1063/1.2987711.

(4) Supady, A.; Blum, V.; Baldauf, C. First-Principles Molecular Structure Search with a Genetic Algorithm. *J. Chem. Inf. Model.* **2015**, *55*, 2338–2348, DOI: 10.1021/acs.jcim.5b00243.

(5) Reymond, J. L. The Chemical Space Project. *Acc. Chem. Res.* **2015**, *48*, 722–730, DOI: 10.1021/ar500432k.

(6) Unterthiner, T.; Mayr, A.; Klambauer, G.; Hochreiter, S. Toxicity Prediction using Deep Learning. *arXiv* **2015**, DOI: 10.3389/fenvs.2015.00080.

(7) Wang, Y.; Zeng, J. Predicting drug-target interactions using restricted Boltzmann machines. Bioinformatics. 2013; pp i126–i134, DOI: 10.1093/bioinformatics/btt234.

(8) Hachmann, J. et al. Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry – the Harvard Clean Energy Project. *Energy Environ. Sci.* **2014**, *7*, 698–704, DOI: 10.1039/C3EE42756K.

(9) Kanal, I. Y.; Owens, S. G.; Bechtel, J. S.; Hutchison, G. R. Efficient computational screening of organic polymer photovoltaics. *J. Phys. Chem. Lett.* **2013**, *4*, 1613–1623, DOI: 10.1021/jz400215j.

(10) Gómez-Bombarelli, R. et al. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat. Mater.* **2016**, *15*, 1120–1127, DOI: 10.1038/nmat4717.

(11) Lin, K.; Gómez-Bombarelli, R.; Beh, E. S.; Tong, L.; Chen, Q.; Valle, A.; Aspuru-Guzik, A.; Aziz, M. J.; Gordon, R. G. A redox-flow battery with an alloxazine-based organic electrolyte. *Nat. Energy* **2016**, *1*, 16102, DOI: 10.1038/nenergy.2016.102.

(12) Huskinson, B.; Marshak, M. P.; Suh, C.; Er, S.; Gerhardt, M. R.; Galvin, C. J.; Chen, X.; Aspuru-Guzik, A.; Gordon, R. G.; Aziz, M. J. A metal-free organic–inorganic aqueous flow battery. *Nature* **2014**, *505*, 195–198, DOI: 10.1038/nature12909.

(13) Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. **2014**,

(14) Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. *NIPS* **2016**, 1–10, DOI: arXiv:1504.01391.

(15) Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *arXiv* **2016**, DOI: 10.1001/jamainternmed.2016.8245.

(16) Jaques, N.; Gu, S.; Bahdanau, D.; Miguel, J.; Lobato, H.; Turner, R. E.; Eck, D. Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control. *arXiv* **2017**,

(17) Ranzato, M.; Chopra, S.; Auli, M.; Zaremba, W. Sequence Level Training with Recurrent Neural Networks. *Iclr* **2016**, 1–15, DOI: 10.1371/journal.pcbi.1005055.

(18) Kusner, M. J.; Hernández-Lobato, J. M. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. *Arxiv* **2016**, 1–6, DOI: 10.1016/j.physa.2016.11.004.

(19) Narasimhan, K.; Kulkarni, T.; Barzilay, R. Language Understanding for Text-based Games Using Deep Reinforcement Learning. *Arxiv* **2015**, 1–11, DOI: 10.18653/v1/D15-1001.

(20) Li, J.; Monroe, W.; Ritter, A.; Jurafsky, D. Deep Reinforcement Learning for Dialogue Generation. *arXiv* **2016**, *2*, 1192–1202, DOI: 10.1103/PhysRevB.94.020410.

(21) Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489, DOI: 10.1038/nature16961.

(22) Gómez-Bombarelli, R.; Duvenaud, D.; Hernández-Lobato, J. M.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *arXiv ID 1610.02415* **2016**, 1–23.

(23) Kadurin, A.; Nikolenko, S.; Khrabrov, K.; Aliper, A.; Zhavoronkov, A. druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Mol. Pharm.* **2017**, acs.molpharmaceut.7b00346, DOI: 10.1021/acs.molpharmaceut.7b00346.

(24) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. **2017**,

(25) Guimaraes, G. L.; Sanchez-Lengeling, B.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. **2017**, DOI: 1705.10843.

(26) James, C. A.; Apodaca, R.; O'Boyle, N.; Dalke, A.; van Drie, J.; Ertl, P.; Hutchison, G.; Landrum, G.; Morley, C.; Willighagen, E.; Winter, H. D.; Vandermeersch, T.; May, J. OpenSMILES specification. **2016**,

(27) O'Boyle, N. M. Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *J. Cheminform.* **2012**, *4*, 22, DOI: 10.1186/1758-2946-4-22.

(28) Landrum, G. RDKit - Open-Source Cheminformatics. 2017; http://www.rdkit.org/.

(29) Kim, Y. Convolutional Neural Networks for Sentence Classification. *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. (EMNLP 2014)* **2014**, 1746–1751, DOI: 10.1109/LSP.2014.2325781.

(30) Hochreiter, Sepp and Schmidhuber, J.; Urgen, Long Short Term Memory. *Memory* **1996**, *9*, 1–28, DOI: 10.1.1.56.7752.

(31) Williams, R. J. Simple Statistical Gradient-Estimating Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* **1992**, *8*, 229–256.

(32) Wildman, S.; Crippen, G. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Model.* **1999**, *39*, 868–873, DOI: 10.1021/ci9903071.

(33) Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **2009**, *1*, 8, DOI: 10.1186/1758-2946-1-8.

(34) Ertl, P.; Roggo, S.; Schuffenhauer, A. Natural Product-likeness Score and Its Application for Prioritization of Compound Libraries. *J. Chem. Inf. Model.* **2008**, *48*, 68–74, DOI: 10.1021/ci700286x.

(35) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **2012**, *4*, 90–98, DOI: 10.1038/nchem.1243.

(36) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. data* **2014**, *1*, 140022, DOI: 10.1038/sdata.2014.22.

(37) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864–2875, DOI: 10.1021/ci300415d.

(38) Irwin, J. J.; Shoichet, B. K. ZINC - A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model* **2005**, *45*, 177–182, DOI: 10.1021/ci049714.

(39) Sterling, T.; Irwin, J. J. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324–2337, DOI: 10.1021/acs.jcim.5b00559.

(40) Jean-Claude, B.; Andrew, L.; Antony, W. Jean-Claude Bradley Double Plus Good (Highly Curated and Validated) Melting Point Dataset. 2014.

(41) Lipinski, C.; Lombardo, F.; Dominy, B.; Feeney, P. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Develop ment Settings. *Adv. Drug Deliv. Rev.* **1997**, *23*, 3–25, DOI: 10.1016/S0169-409X(00)00129-0.

(42) Law, V. et al. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Res.* **2014**, *42*, D1091–D1097, DOI: 10.1093/nar/gkt1068.

(43) Drefahl, A. CurlySMILES: a chemical language to customize and annotate encodings of molecular and nanodevice structures. *J. Cheminform.* **2011**, *3*, 1, DOI: 10.1186/1758-2946-3-1.

(44) Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. Grammar Variational Autoencoder. **2017**, DOI: 1703.01925.

(45) Rajeswar, S.; Subramanian, S.; Dutil, F.; Pal, C.; Courville, A. Adversarial Generation of Natural Language. *arXiv* **2017**, DOI: 1705.10929.

(46) Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; Xing, E. P. Controllable Text Generation. *arXiv* **2017**, DOI: 1703.00955.