Deep molecular dreaming: inverse machine learning for *de-novo* molecular design and interpretability with surjective representations

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 03LT02 https://doi.org/10.1088/2632-2153/ac09d6

MACHINE LEARNING
Science and Technology

LETTER

Deep molecular dreaming: inverse machine learning for *de-novo* molecular design and interpretability with surjective representations

Cynthia Shen[1,5,*], **Mario Krenn**[1,2,3,5,*], **Sagi Eppel**[1,2,3,*] **and Alán Aspuru-Guzik**[1,2,3,4,*]

[1] Department of Computer Science, University of Toronto, Toronto, Canada
[2] Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, Canada
[3] Vector Institute for Artificial Intelligence, Toronto, Canada
[4] Canadian Institute for Advanced Research (CIFAR) Lebovic Fellow, Toronto, Canada
[5] These authors contributed equally.
[*] Authors to whom any correspondence should be addressed.

E-mail: cynt.shen@mail.utoronto.ca, mario.krenn@utoronto.ca, sagieppel@gmail.com **and** alan@aspuru.com

Keywords: inverse design, DeepDream, inceptionism, cheminformatics, SELFIES, de-novo molecular design, deep generative model

OPEN ACCESS

RECEIVED
20 February 2021

REVISED
6 May 2021

ACCEPTED FOR PUBLICATION
9 June 2021

PUBLISHED
13 July 2021

Abstract

Computer-based *de-novo* design of functional molecules is one of the most prominent challenges in cheminformatics today. As a result, generative and evolutionary inverse designs from the field of artificial intelligence have emerged at a rapid pace, with aims to optimize molecules for a particular chemical property. These models 'indirectly' explore the chemical space; by learning latent spaces, policies, and distributions, or by applying mutations on populations of molecules. However, the recent development of the SELFIES (Krenn 2020 *Mach. Learn.: Sci. Technol.* **1** 045024) string representation of molecules, a surjective alternative to SMILES, have made possible other potential techniques. Based on SELFIES, we therefore propose PASITHEA, a direct gradient-based molecule optimization that applies inceptionism (Mordvintsev 2015) techniques from computer vision. PASITHEA exploits the use of gradients by directly reversing the learning process of a neural network, which is trained to predict real-valued chemical properties. Effectively, this forms an inverse regression model, which is capable of generating molecular variants optimized for a certain property. Although our results are preliminary, we observe a shift in distribution of a chosen property during inverse-training, a clear indication of PASITHEA's viability. A striking property of inceptionism is that we can directly probe the model's *understanding* of the chemical space on which it is trained. We expect that extending PASITHEA to larger datasets, molecules and more complex properties will lead to advances in the design of new functional molecules as well as the interpretation and explanation of machine learning models.

1. Introduction

The *de-novo* design of new functional chemical compounds can bring enormous scientific and technological advances. For this reason, researchers in cheminformatics have developed a plethora of AI methodologies for the challenging inverse molecular design task [3, 4]. They include deep learning techniques such as variational autoencoders (VAE) [5–7], generative adversarial networks (GAN) [8, 9], reinforcement learning (RL) [10, 11], and evolutionary techniques such as genetic algorithms (GA) [12–15].

These methods belong to a category with one particular attribute: the model *indirectly* optimizes molecules for a target property. For example, VAEs and GANs learn to mimic a distribution of molecules from a training set, constructing a latent space that is then scanned to find molecules that optimize an objective function. In the case of RL, the agent learns from rewards in the environment in order to build a policy for generating molecules, which is subsequently used to maximize an objective function. Finally, in GAs, the population is optimized iteratively by applying mutations and selections. In all of these cases, the optimization process does not directly maximize the objective function in a gradient-based way.

Here, we present preliminary results for PASITHEA[6], a new generative model for molecules inspired by inceptionism techniques [16] in computer vision. PASITHEA is a gradient-based method that optimizes a discrete molecular structure for a target property. We train a neural network to predict chemical properties using a molecular string representation. We then invert the training of the network to generate new variants of molecules. This is a largely stochastic process that produces a pool of molecules that is considerably more diverse than the original dataset. This approach has two significant novelties:

- Molecules are *directly* optimized to a given objective function, sidestepping the learning of distributions and policies, or the application of mutations to a population.
- We can analyse what the regression network has learned about the chemical property by probing its inverse training with test molecules. This may allow us to explain the neural network's understanding of chemistry.

Furthermore, in contrast to most exploratory methods such as RLs or GAs, PASITHEA does not require expensive function evaluations for quantum chemistry calculations. Provided that we use a pre-calculated dataset, this is an important advantage over explorative approaches such as GA or RL, since costly chemical properties can be directly optimized.

This method is made possible by the application of SELFIES, a 100% robust molecular string representation [17]. In contrast to SMILES, for which a large fraction of generable strings do not map to valid molecular graphs, SELFIES is a surjective map between molecular strings and molecular graphs. That is, for every SELFIES string, there exists a valid molecular graph, and every molecular graph can be represented by SELFIES.

In this analysis, the logarithm of partition coefficient (logP) is used as our target property, obtained from the RDKit library [18]. The logP, which measures the lipophilicity of a molecule, is an important property of drug molecules and an indicator of drug-likeness [19]. Apart from its practicality, the logP is particularly suitable to this study because it follows an approximate normal distribution in the QM9 dataset. The range of logP values is nearly continuous, which is essential for gradient-based molecular design.

Once PASITHEA has been trained on the QM9 dataset to predict logP values, we initialize the inverse training with a molecule and optimize it for a logP value. We confirm a shift in the logP distribution of generated molecules. We can observe how the model changes a molecule quasi-continuously over several steps to a final, optimized chemical structure. Finally, we indicate how this technique can be used to probe concepts learned by PASITHEA. For completeness, we want to mention there are very recent technologies that can efficiently explore the chemical space that do not require training, datasets or domain-knowledge at all [20]. Those techniques are complementary to what we demonstrate here, and it might be interesting to see how they can be combined with PASITHEA.

2. Methodology

2.1. Inceptionism

Despite wide-ranging applications, artificial neural networks are still a black box—there is a wealth of research in machine learning devoted to understanding how a network learns from data. Particularly, inceptionism has drawn considerable attention as a technique to visualize the inner workings of a network, capable of designing distinctive works of art [1, 16]. At the core of this technique is the idea that after a classification network has been trained on a dataset of images, each layer encodes some feature or aspect that allows it to distinguish between images. Deeper layers hold high-level features such as the shapes of specific objects; layers closest to the input tend to hold simple aspects such as texture. In a process referred to as 'deep dreaming', the initial classification training is reversed. Here, the network is no longer updating the weights and biases by minimizing the error between prediction and ground truth. Rather, it is maximizing the activations of a chosen layer. A single image chosen as input is then gradually mutated via gradient ascent as the layer is enhanced, displaying a visual interpretation of the layer. An example is shown in figure 1, where animal features are enhanced in the image of a chemistry lab. The final image shown in figure 1 is generated by holding the network weights fixed while modifying the pixel values of the first image, maximizing the output of specific internal neurons. In this way, by looking at the final picture, we can infer the role of these specific neurons in the entire neural network.

We generalize this technique to the inverse-design task of functional molecules. Several key modifications to the original algorithm are required. The network is no longer predicting discrete image classes, but a

[6] PASITHEA is the goddess of relaxation, meditation, hallucinations, and wife of Hypnos, the god of sleep. The full code is available at https://github.com/aspuru-guzik-group/Pasithea.

Figure 1. Inceptionism techniques aim to understand the inner mechanism of a neural network by visualizing the information encoded in a particular layer [1]. A typical result after training on a chosen image is a build-up of flourishes that has a dream-like quality—hence the name of the process, 'dreaming'. To generate this image, we use the following github repository [2].

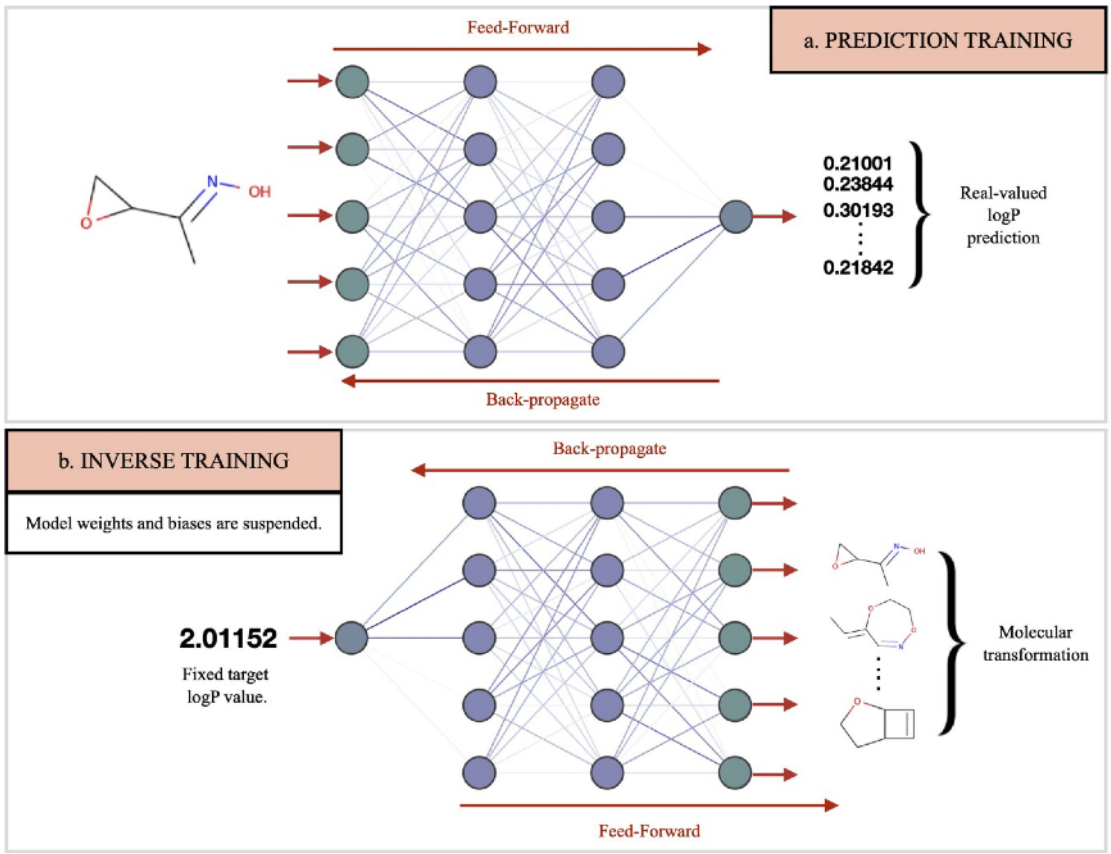


Figure 2. Two-step training for PASITHEA. (a) In prediction training, the neural network learns to predict logP values from input molecules through the standard feedforward and backpropagation process in which the network weights are updated continually. (b) The same feedforward and backpropagation process occurs. The gradients are not computed with respect to the weights, but rather the input molecule.

continuous property value such as the logP. Moreover, the inverse training does not deal with any arbitrary layer. Instead, we are most interested in the output layer, which contains the predicted property value.

Prior to deep dreaming, the network learns to predict a specific real-valued property for each molecule in a given dataset (i.e. logP) from the molecular graph. The training involves the standard feedforward and backpropagation process. For a set of fixed inputs and outputs, the network iteratively improves its predictions by updating the weights through mini-batch gradient descent (figure 2(a)). In deep dreaming, an input molecule with a property value predicted by the network is incrementally modified to a similar molecule with the desired value. The weights and biases of each layer of the network are now fixed and the neural network is no longer adjusting its logP prediction for each molecule. Through backpropagation, we minimize the error between the predicted properties of each input molecule and the desired target property (figure 2(b)). The computed error is then used to compute the gradient with respect to the one-hot encoding of the input. This effectively transforms the input gradually to a molecule that matches the target property. Each increment of the one-hot encoding corresponds to a potential transformation of the input molecule. Once the loss function has been minimized, the gradient evaluates approximately to zero, which terminates

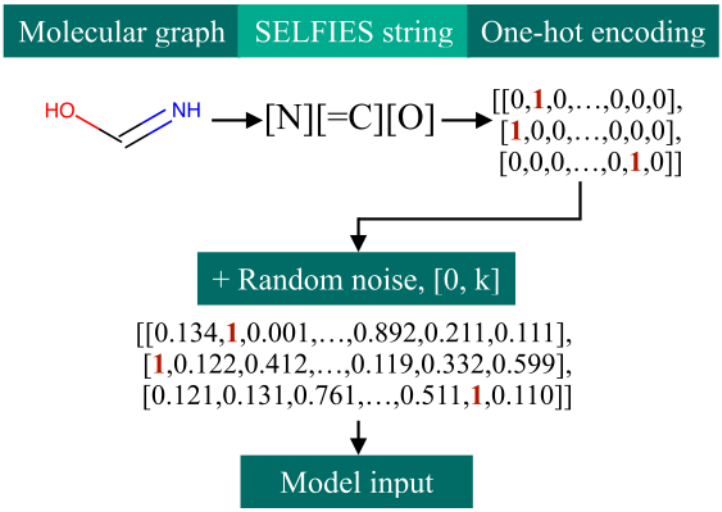


Figure 3. It is an ongoing study to find the optimal method to convert a SELFIES string into an input suitable for deep dreaming. As described in the original paper [17], each molecule is tokenized into a set number of SELFIES symbols and processed into one-hot encoded vectors. The number of symbols is determined by the length of the largest SELFIES string in the dataset. In tokenizing molecules which require less symbols than the set number, we use a special null token for the empty positions. When a one-hot encoding is taken as input, the encoding transforms from a vector of binary variables to a vector of real numbers through the first and subsequent iterations of inverse training. The mixed vector representations make it difficult for the deep dreaming model to converge. Therefore, taking one of several possible approaches, we introduce noise in the one-hot encoding as input to deep dreaming. Every zero in the one-hot encoding is replaced by a random number in the interval $[0, k]$, where the upper-bound k is set to a value between 0.5 and 0.95.

the training. In this process, the same standard feedforward and backpropagation algorithm is used, but the input molecule is adjusted while the weights and biases remain constant.

2.2. Molecular representation

An important contribution to the model is a recently developed textual representation for molecules known as SELFIES. The dreaming process requires a continuous space in which all points are valid, a criterion met by SELFIES, which has been proven to be 100% valid [17]. The traditional SMILES representation can be problematic when the deep dreaming model transitions over an invalid structure from one molecule to the next. For example, in the transition from a string containing a ring, 'CCCC1CCCC1CC', to a string without a ring, 'CCCCCCCCCC', the model is likely to produce strings resembling 'CCCC1CCCCCC', which does not correspond to a valid molecular graph. In this case, the transformation may reveal the network's understanding of string syntax in relation to logP, but not the molecular structure in relation to logP, since the string does not correspond to a valid molecule. In contrast, the SELFIES representation enforces a constraint on the syntax to prevent the model from producing such invalid structures, which produces a complete optimization sequence that directly maps to valid molecules.

There are several steps required to convert from a SELFIES string to a one-hot encoded vector for the model input, described in figure 3. The last step, applying noise to input vectors, ensures that we observe increments in the molecular optimization, instead of haphazard modifications or none at all.

A more straightforward task is the opposite conversion, from the one-hot encoding to the corresponding SELFIES string, in each iteration of inverse training. We apply the argmax function to each array to find the position of the maximum value, which is then translated to the corresponding SELFIES symbol.

3. Comparison to VAEs

A simple four-layer network highlights one key difference between PASITHEA and other optimization methods: we perform reverse-differentiation directly on the molecular representation, which is a one-hot encoding of SELFIES. Let us compare this approach with the related concept of VAE [5]. In VAEs, a latent space is learned by encoding and decoding molecules. After the reconstruction, another neural network can then optimize in the newly created latent space. In this case, since the prediction network is applied to the latent space, the basis for gradient computation lies in the latent space, not in the molecular representation itself.

The direct reversibility on the basis of model weights is important in the context of machine learning interpretability. Our goal is to understand directly what a neural network learns about a specific molecular

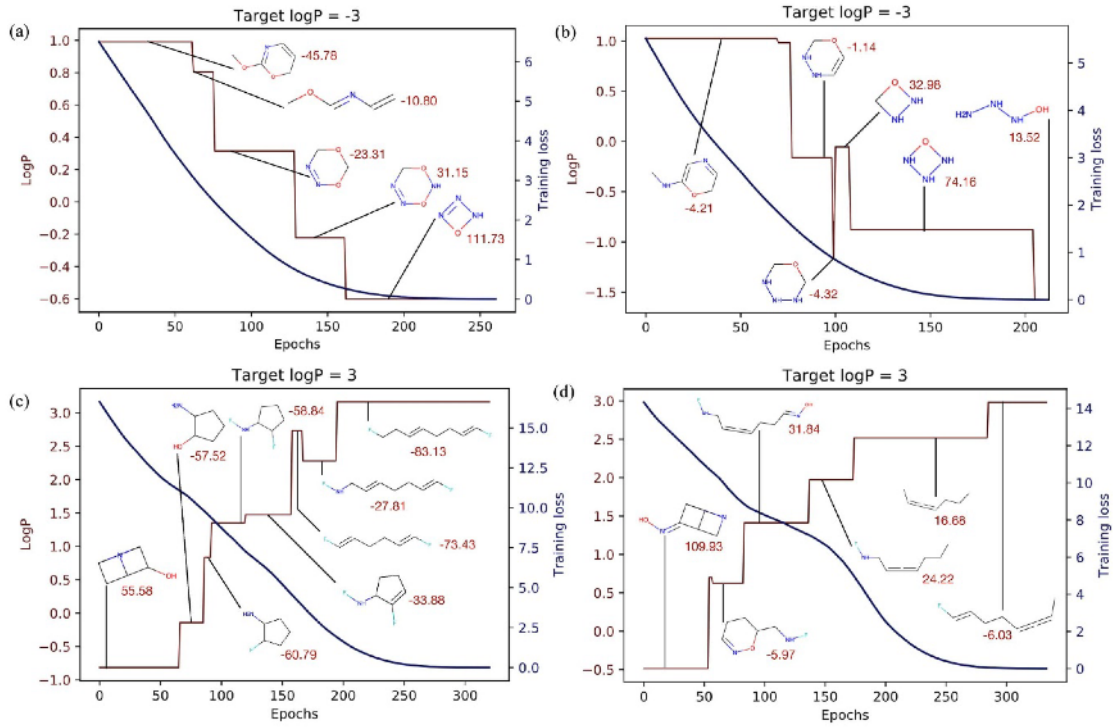


Figure 4. Stepwise molecule transformations, optimized for a lower target logP in (a), (b) and optimized for a higher target logP in (c), (d). The blue curve represents the training loss over the number of epochs and the red curve represents logP adjustments corresponding to the training loss, where the logP values are computed with RDKit. Next to each molecular structure is a number written in red, indicating the computed heats of formation in kcal/mol. In some cases, the molecule evolves to meet the target by adding a noticeable pattern repeatedly, such as the appendage of nitrogen in (a) and length reduction in (b). Figures (c) and (d) show the analogous behaviour for an optimization target with a large positive logP. Notice, in (b), the significant aberration at around epoch 100. The logP adjustment is initially done in the correct direction, but it is immediately snapped back in the opposite direction. This is a case where the neural network over-generalizes and does not capture the full relationship between molecular representation and logP.

property. We believe that probing the regression neural network with test molecules, without a detour over some specific latent spaces, is the most direct way to understand what the model has learned.

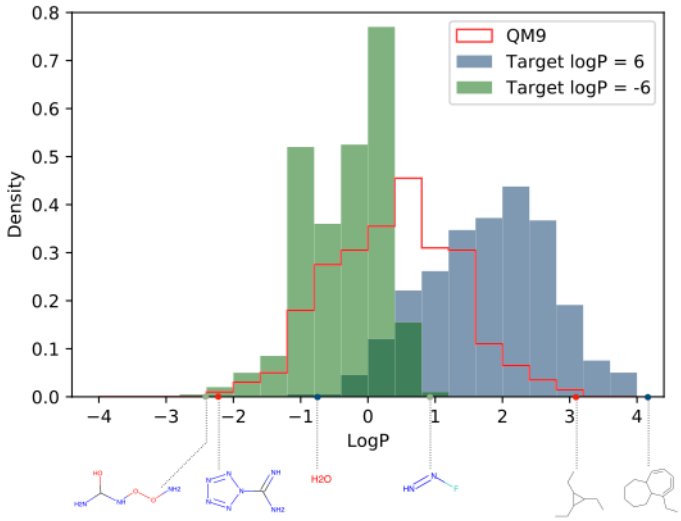
4. Results

A simple four-layer neural network, with no added components, suffices for our results and we did not require an exhaustive search for the ideal training hyperparameters. A random subset of 10 000 molecules from the QM9 dataset is selected as input to a network trained to predict logP values. We demonstrate how PASITHEA transforms molecules in a stepwise, quasi-continuous fashion and shifts the distribution of logP in the molecular dataset toward set targets. These logP targets are set high in order to observe a rightward shift in logP distribution and similarly set low to observe a shift in the opposite direction. With logP targets much further from the central tendency in the distribution, surpassing the highest and lowest values in the dataset, we observe a more pronounced shift in distribution during training. We then analyse what PASITHEA has learned regarding the relationship between logP and molecular structure. Our experiments clearly indicate that deep dreaming achieves both a direct, gradient-based design of novel functional molecules and the explainability of neural networks for molecules.

4.1. Evolution of individual molecules

Of particular interest is the gradual progression of each molecule through inverse training. Over hundreds of training epochs, the gradient with respect to input SELFIES produces minor adjustments in the molecule that increments to a pronounced transmutation (figure 4). The behaviour of these adjustments are stepwise due to the discrete, textual nature of the molecules represented by strings, but continuous in terms of real-valued one-hot encodings.

To test the quality of generated molecules, we consider thermodynamic stability as a metric of synthesizability. Next to each molecular structure in figure 4 is a number coloured in red, indicating the computed heats of formation in kcal mol^{-1}. Smaller, negative numbers, as shown in figure 4(c), mark a progression toward high thermodynamic stability. In figures 4(a), (b) and (d), we observe much larger,



	QM9 molecules	Optimized molecules (target +6)	Optimized molecules (target -6)
Mean logP	0.3909	1.8172	-0.3360
Minimum logP	-2.1903	-0.8240	**-2.452**
Maximum logP	3.0786	**4.2442**	0.9018

Figure 5. The molecules are optimized toward logP targets at both extreme ends, which indicates a non-trivial understanding of structure–property relationships. Depicted underneath the histogram are the molecules corresponding to the maximum and minimum logP values from each distribution. LogP values shown here have been computed with RDKit. Notice that the right tail and the left tail of the shifted distributions (in blue and green) extend past those of the original distribution (highlighted in red).

positive numbers, indicating low stability. Although the heats of formation is a useful measure, we expect that it is not optimized well in the inverse training. In order to generate molecules that are more stable, for instance, the heats of formation can be incorporated as a target in prediction training, as a multi-objective optimization. Using inceptionism, multi-target optimization works analogously to the single-objective optimization.

4.2. Shift in distribution

In order to observe a large-scale pattern over the entire dataset, we disregard the intermediate molecules and restrict our analysis to the initial and fully-optimized molecules. We take a sample of 10 000 random molecules from the QM9 dataset and apply deep dreaming to each molecule. From these results, there is a clear shift in the distribution of logP values in the set of molecules as they transmute toward a given target value (figure 5). The distribution shifts in figure 5 show that there are some molecules generated with logP values exceeding the lowest and highest values in the original dataset. For instance, notice that the left tail of the left-shifted (green) distribution extends beyond the left tail of the original (red) distribution and the right tail of the right-shifted (blue) distribution extends beyond the right tail of the original distribution. A quantitative account is summarized under the histogram in figure 5. Notice that the maximum logP in the right-shifted distribution exceeds that in the original dataset, and the minimum in the original dataset exceeds that in the left-shifted distribution. Demonstrably, PASITHEA is generating novel molecules with properties outside the limits of the original training set of molecules, which attests to the large potential of this method.

4.3. Probing the neural network's intuitions

4.3.1. Interpretable ML in physics

Our approach to *de-novo* molecular generation does not require domain knowledge, nor is the design of PASITHEA influenced by domain knowledge. However, this knowledge is useful when applied to individual molecular evolutions in deep dreaming [21]. In particular, we take interest in the recent progress in the machine-assisted discovery of concepts in the natural sciences [22–27]. These lines of research use machine learning techniques to draw conclusions about the underlying processes of a particular physical system, which are often mathematical models with tunable parameters that are responsive to input observations. This approach differs from research in machine-assisted *de-novo* molecular generation [5], where the focus lies in

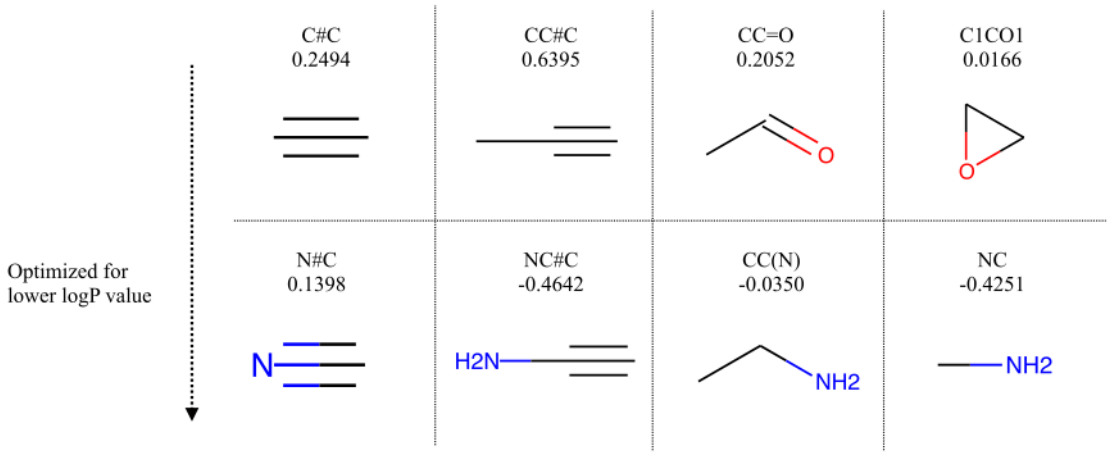


Figure 6. Graphical and SMILES representation of nitrogen appendages, with the corresponding logP. LogP values shown here have been computed with RDKit. Shown in the top row are initial test molecules from the QM9. Shown in the bottom row are the optimized counterparts of the molecules directly above. The neural network recognizes the nitrogen atom as an important indicator of a low logP value.

producing optimization methods that can navigate a massive chemical search space. Our approach may close the gap between these lines of research. By inverting the training, we achieve both molecular generation and insights into how the network produces each molecular transformation, such as the 'strategy' employed to optimize molecules by appending nitrogen atoms. Although PASITHEA does not model the behaviour of molecules in the physical sense, it does model the transition rules required for molecular optimization; there is potential in rigorously quantifying these transition rules. Specifically, the viability of inceptionism in recovering the thermodynamic principles of physics [27] attests to the potential for chemistry.

4.3.2. Interpretable ML in chemistry

Inspired by explainable representations in image recognition [28] and rediscovery of concepts in physics [27], we can *understand the internal molecular representation by inverting it*. For that, we probe the neural network with specific test molecules and observe patterns in how it changes them. For example, the composition of atoms after inverse training follows a predictable pattern, such as the appendage of a few non-carbon atoms, fluorine and nitrogen. Take, for example, the transmutations of the simplest molecules in the QM9 dataset (figure 6), which suggest that PASITHEA interprets these non-carbon atoms as correlated with lower logP values. A similar trend persists for more complex molecules, in which more than one atom may be replaced with nitrogen (figure 4(a)), though this persists to a lower extent for fluorine.

The intermediate states during continuous transformation can be used as additional insights into the network's understanding of chemical property. In particular, by observing a single test molecule, there are instances where an additional iteration in inverse-training transforms the molecule with a repeated 'strategy' that has been used in previous iterations. The neural network appears to persist with a single strategy until the training terminates.

We demonstrate this behaviour in figure 4(b), which shows a gradual process of reducing length, and in figure 4(a), which shows an initial molecule containing a single nitrogen atom, an intermediate molecule containing two nitrogen atoms, and a final molecule containing three nitrogen atoms. Moreover, in figure 7(c), the neural network persists with several strategies. In the first three transformations, we observe that the network replaces a single bond with a double bond each time, growing from a total of one double bond to three double bonds. In the last two transformations, the network adds another double bond, while in the process replacing a bond with a triple bond two times.

We discern another possible trend in figure 7, though less apparent, by noticing differences between the molecules generated in positive and negative optimization. The generated molecules are shown to be reducing in length in figure 7(a) and (b), while in figures 7(c) and (d), this pattern is either absent or the structure appears to be increasing in length. The network has some non-trivial understanding of molecular length: in prediction training, it has discovered that smaller molecules tend to associate with a lower logP value. Similarly, figure 7(a) is a clearer indication that the network understands molecular length, taking a simple five-carbon molecule as input. When optimizing in one direction, the network adds carbon to the molecule, and when optimizing in the opposite direction, the network removes carbon several times. Put

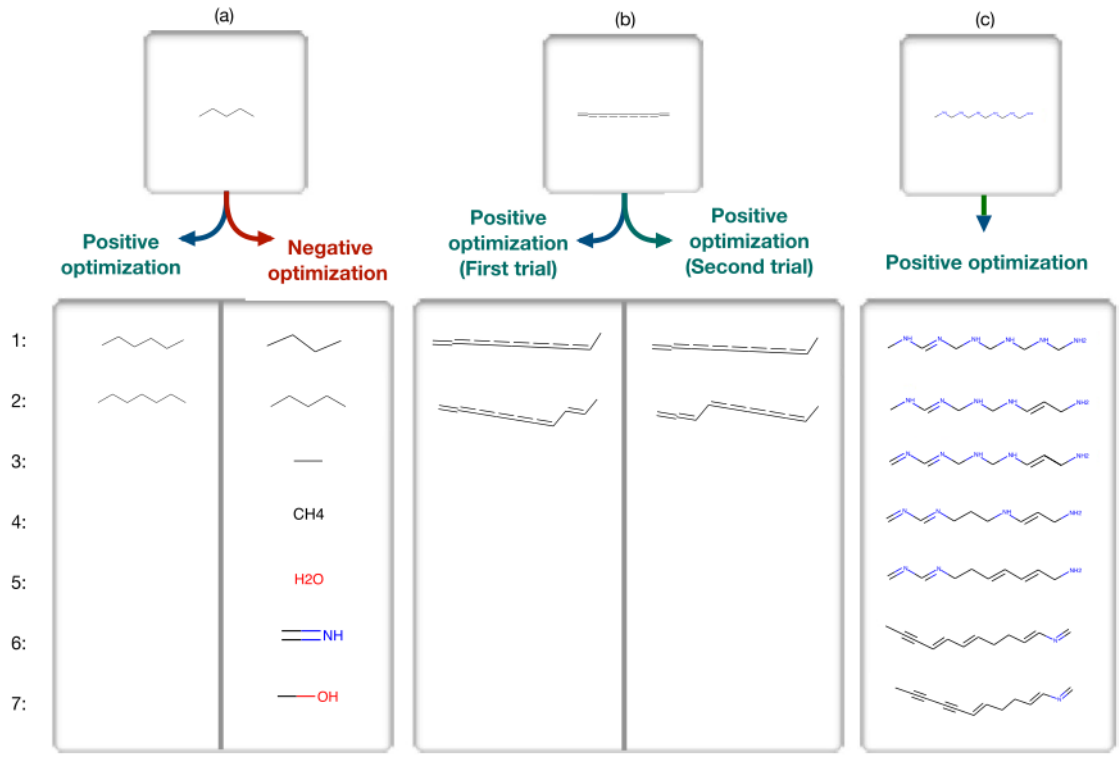


Figure 7. Probing the intermediate molecules. In each case, a positive optimization sets the logP value $\log P_{\text{target}} = +6$ as the target, and a negative optimization sets $\log P_{\text{target}} = -6$ as the target. The neural network persists with certain 'strategies' in making the molecular transformations: (a) in optimizing to the +6 target, the network consistently increases the length of the string of carbons. The opposite occurs for the −6 target. (b) Double bonds are repeatedly replaced by a carbon atom, which occurs over multiple trials of positive optimization. (c) A single bond is repeatedly replaced by a double bond or triple bond when optimizing to the +6 target.

differently, the network not only optimizes toward a goal arbitrarily—it recognizes dichotomies in structural features and exploits them selectively.

As a hallmark of the dreaming, we find that 97.2% of the molecules in the generated set do not exist in the original training set. The network is not memorizing the data, instead using features of the molecular structure to guide its optimization. For instance, the number of heavy atoms in some generated molecules exceed the maximum number, 9, in the original QM9 dataset. The only limit to the size of any generated molecule is governed by the string representation, SELFIES—more specifically, the set number of symbols used to express each molecule. Since the length of the largest SELFIES string in the QM9 is larger than 20, the set number of symbols is larger than 20. Therefore, the deep dreaming model could potentially replace every symbol with a heavy atom, generating a molecule containing more than 20 heavy atoms.

Finally, an inspection of repeated trials of the same optimization process gives some more intuition into the network's understanding. Since the input one-hot encoding is initialized with random noise before inverse-training, the transformation is entirely stochastic. As a result, repeated trials of the same molecule give different sequences of transformation. Despite this factor, the network demonstrates consistent understanding across several trials. Some examples are shown in figure 7. Repeated trials for the molecule in figure 7(a) may not result in an identical sequence, but a similar extension and reduction process occurs. Figure 7(b) shows two selected trials for another test molecule, and in both trials some double-bonds are persistently replaced with carbon atoms.

These cases validate that the network is charting deliberate, non-arbitrary paths toward the target logP; it has a non-trivial understanding of features corresponding to higher and lower logP values.

5. Outlook

We propose a direct, gradient-based property-optimization method that offers insights into the network's understanding of structure–property relationships. Our next task is to explore other 'strategies' the network may use in optimizing for other chemical properties, including those that require expensive quantum chemistry calculations. Moreover, from a more practical standpoint, the quality of generated molecules is

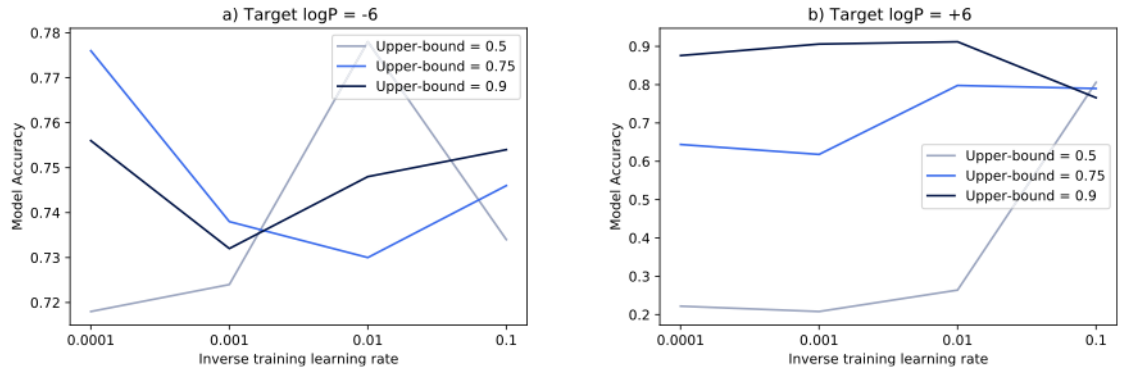


Figure 8. Hyperparameter tuning for inverse training.

another important consideration. A multi-objective design would allow more than one property to be optimized in the inverse training, including a metric for synthesizability such as heats of formation.

There is also work to be done to add transparency [21] to our approach. There are many possible directions, including exploring other surjective string representations that may be more suitable to the task of deep dreaming, and comparing other reverse-differentiable machine learning architectures that may be capable of a similar 'dreaming' process. Ultimately, our work can be used to find the underlying rules the neural network discovers in order to optimize a property, conjointly offering insights into how the network makes its predictions for interpretability and suggesting ways in which a human can use these rules in order to generate new and useful chemical compounds for explainability [21].

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files). The full code, along with the model, is available at https://github.com/aspuru-guzik-group/Pasithea.

Acknowledgments

The authors thank Robert Pollice for help with chemistry. A A-G acknowledges generous support from the Canada 150 Research Chair Program, Tata Steel, Anders G Froseth, and the Office of Naval Research. M K acknowledges support from the Austrian Science Fund (FWF) through the Erwin Schrödinger Fellowship No. J4309.

Appendix A. Neural network architecture and training

PASITHEA uses a fully-connected neural network consisting of four layers, each with 500 nodes, using the reLU activation function. The loss function during training is determined by computing the mean squared error. Throughout prediction training, the data is split into two parts: 85% and 15% of the data for training and testing the model, respectively. We train the prediction model for about 1500 epochs. Both the prediction training and inverse training use an Adam optimizer.

An important hyperparameter is the upper-bound on the level of noise applied to the input vectors. We find an upper-bound of 0.95 appropriate for prediction training. For the current network architecture, the learning rate must be set low for prediction training, at 1×10^{-6}. Holding these hyperparameters fixed, a summary of varying upper-bounds and learning rates for the inverse training is listed in figure 8. The model accuracy is evaluated by the percent of molecules optimized toward the target property. From this data, we choose 0.9 as the upper-bound and 0.01 as the learning rate for the inverse training.

Appendix B. Heats of formation

Heats of formation (displayed in figure 4 as an indicator of stability) were estimated using MOPAC2016 [29] at the PM7 [30] level of theory with PRECISE settings by performing geometry optimization of the initial

guess structures using eigenvector following (keyword EF). Initial guess structures were generated from the SMILES strings using Molconvert [31].

ORCID iDs

Cynthia Shen https://orcid.org/0000-0003-3266-2634
Mario Krenn https://orcid.org/0000-0003-1620-9207
Alán Aspuru-Guzik https://orcid.org/0000-0002-8277-4434

References

[1] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition (arXiv:1409.1556)
[2] Linder-Noren E 2019 Pytorch—deep dream (available at: https://github.com/eriklindernoren/PyTorch-Deep-Dream)
[3] Sanchez-Lengeling B and Aspuru-Guzik Aan 2018 Inverse molecular design using machine learning: generative models for matter engineering *Science* **361** 360–5
[4] Coley C W 2021 Defining and exploring chemical spaces *Trends in Chemistry* **3** 133–45
[5] Gómez-Bombarelli R *et al* 2018 Automatic chemical design using a data-driven continuous representation of molecules *ACS Central Sci.* **4** 268–76
[6] Jin W, Barzilay R, and Jaakkola T 2018 Junction tree variational autoencoder for molecular graph generation (arXiv:1802.04364)
[7] Tengfei M, Chen J and Xiao C 2018 Constrained generation of semantically valid graphs via regularizing variational autoencoders (arXiv:1809.02630)
[8] 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:1705.10843)
[9] Nicola D C and Kipf T 2018 Molgan: an implicit generative model for small molecular graphs (arXiv:1805.11973)
[10] Zhou Z, Kearnes S, Li Li, Zare R N and Riley P 2019 Optimization of molecules via deep reinforcement learning *Sci. Rep.* **9** 1–10
[11] You J, Liu B, Ying Z, Pande V and Leskovec J 2018 *Advances in Neural Information Processing Systems* pp 6410–21
[12] Jensen J H 2019 A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space *Chem. Sci.* **10** 3567–72
[13] Nigam A, Friederich P, Krenn M and Aspuru-Guzik Aan 2019 Augmenting genetic algorithms with deep neural networks for exploring the chemical space (arXiv:1909.11655)
[14] Henault E S, Rasmussen M H and Jensen J H 2020 Chemical space exploration: how genetic algorithms find the needle in the haystack *PeerJ. Phys. Chem.* **2** e11
[15] Reeves S, DiFrancesco B, Shahani V, MacKinnon S, Windemuth A and Brereton A E 2020 Assessing methods and obstacles in chemical space exploration *Appl. AI Lett.* **1** e17
[16] Mordvintsev A, Olah C and Tyka M 2015 Inceptionism: going deeper into neural networks (available at: https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html)
[17] Krenn M, Hase F, Nigam A, Friederich P and Aspuru-Guzik A 2020 Self-referencing embedded strings (selfies): a 100% robust molecular string representation *Mach. Learn.: Sci. Technol.* **1** 045024
[18] Landrum G *et al* 2006 Rdkit: open-source cheminformatics
[19] Lipinski C A 2004 Lead-and drug-like compounds: the rule-of-five revolution *Drug Discovery Today: Technol.* **1** 337–41
[20] Nigam A, Pollice R, Krenn M, dos Passos Gomes G and Aspuru-Guzik A 2020 Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies *ChemRxiv* 13383266
[21] Adadi A and Berrada M 2018 Peeking inside the black-box: a survey on explainable artificial intelligence (XAI) *IEEE Access* **6** 52138–60
[22] Iten R, Metger T, Wilming H, Lídia del R and Renner R 2020 Discovering physical concepts with neural networks *Phys. Rev. Lett.* **124** 010508
[23] Roscher R, Bohn B, Duarte M F and Garcke J 2020 Explainable machine learning for scientific insights and discoveries *IEEE Access* **8** 42200–16
[24] Friederich P, Krenn M, Tamblyn I and Aspuru-Guzik A 2020 Scientific intuition inspired by machine learning generated hypotheses (arXiv:2010.14236)
[25] Deng Y, Ren S, Fan K, Malof J M and Padilla W J 2021 Neural-adjoint method for the inverse design of all-dielectric metasurfaces *Opt. Express* **29** 7526–34
[26] Ren S, Padilla W and Malof J 2020 Benchmarking deep inverse models over time, and the neural-adjoint method Advances in Neural Information Processing Systems
[27] Seif A, Hafezi M and Jarzynski C 2021 Machine learning the thermodynamic arrow of time *Nat. Phys.* **17** 105–13
[28] Mahendran A and Vedaldi A 2015 Understanding deep image representations by inverting them *Proc. Conf. on Computer Vision and Pattern Recognition* pp 5188–96
[29] Stewart J J P 2020 MOPAC2016 (version: 20.323l)
[30] Stewart J J P 2013 Optimization of parameters for semiempirical methods VI: more modifications to the NDDO approximations and re-optimization of parameters *J. Mol. Model.* **19** 1–32
[31] ChemAxon 2019 Marvin (version: 19.24.0)