# ChemPix: Automated Recognition of Hand-drawn Hydrocarbon Structures Using Deep Learning

Hayley Weir,[1,2] Keiran Thompson,[1,2] Amelia Woodward,[1] Ben Choi,[3] Augustin Braun,[1] and Todd J. Martínez[1,2,*]

[1]Department of Chemistry, Stanford University, Stanford, CA 94305

[2]SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025

[3]Department of Electrical Engineering, Stanford University, Stanford, CA 94305

**Abstract**

Inputting molecules into chemistry software, such as quantum chemistry packages, currently requires domain expertise, expensive software and/or cumbersome procedures. Leveraging recent breakthroughs in machine learning, we develop ChemPix: an offline, hand-drawn hydrocarbon structure recognition tool designed to remove these barriers. A neural image captioning approach consisting of a convolutional neural network (CNN) encoder and a long short-term memory (LSTM) decoder learned a mapping from photographs of hand-drawn hydrocarbon structures to machine-readable SMILES representations. We generated a large auxiliary training dataset, based on RDKit molecular images, by combining image augmentation, image degradation and background addition. Additionally, a small dataset of ~600 hand-drawn hydrocarbon chemical structures was crowd-sourced using a phone web application. These datasets were used to train the image-to-SMILES neural network with the goal of maximizing the hand-drawn hydrocarbon recognition accuracy. By forming a committee of the trained neural networks, we achieved a nearly 10 percentage point improvement of the molecule recognition accuracy and were able to assign a confidence value for the prediction based on the number of agreeing votes. The top ensemble model achieved a hand-drawn hydrocarbon recognition accuracy of 77% for the first prediction and 86% if the top 3 predictions were considered; in over 50% of cases, the model was at least 97% confident in the prediction, making it a promising tool for real-world use cases.

## Introduction

Artificial intelligence (AI) refers to the introduction of "human intelligence" into artificial machines. Machine learning is a subfield of AI that focuses specifically on the "learning" aspect of the machine's intelligence, removing the need for manually coding rules. Although Rosenblatt proposed the perceptron in the 1950s,[1] it wasn't until the 1990s that machine learning shifted from a knowledge-based to a data-driven approach. A decade later, "deep learning" emerged as subclass of machine learning that employed multilayer neural networks (NNs). The boom of big-data and increasingly powerful computational hardware allowed deep learning algorithms to achieve unprecedented accuracy on a variety of problems. This resulted in much of the AI software used today, such as music/movie recommenders, speech recognition, language translation and email spam filters.

Deep learning algorithms have been adopted by almost every academic field with hopes of solving both novel and age-old problems.[2] The natural sciences have historically relied on the development of theoretical models derived from physically-grounded fundamental equations to explain and/or predict experimental observations. This makes data-driven models an interesting, and often unusual, approach. In quantum chemistry, for example, to calculate the energy of a molecule one would traditionally solve an approximation to the electronic Schrodinger equation. A machine learning approach to this problem, however, might involve inputting a dataset of molecules and their respective energies into a NN, which would learn a mapping between the two.[3-5] The ability to generate accurate models by extracting features directly from data without human input makes machine learning techniques an exciting avenue to explore in all areas of chemistry – from drug discovery and material design to analytical tools and synthesis planning.

Easy-to-use machine learning based tools have the potential to accelerate research and enrich education. Here, we develop a hand-drawn molecule recognition tool to extract a digital representation of the molecule from an image of a hand-drawn hydrocarbon structure. Drawing skeletal chemical structures by hand is a routine task for students and researchers in the chemistry community. Therefore, photographing a hand-drawn chemical structure offers a low-barrier method of entering molecules into software that would normally require time-consuming workflows and domain expertise. Moreover, for the vast majority of the chemistry community, drawing a chemical structure by hand is far less cumbersome than building it with a mouse. The recognition tool could be integrated into a phone application that performs tasks such as quantum

chemistry calculations, database lookups and AI synthesis planning directly from the hand-drawn molecule, extending the ChemVox voice-recognition system we recently developed.[6]

In addition to its potential as a chemical research and education widget, hand-drawn hydrocarbon recognition is an interesting problem from a fundamental science perspective: it serves as a prototypical example of how deep learning can be applied to a well-suited chemical problem. Sourcing a large training dataset for this task is time and resource intensive, a common obstacle encountered in machine learning applications. To address this, we discuss strategies for synthetic data generation and their generalizability to scenarios where there is access to limited real-world data, but abundant similar data.

Hand-drawn chemical structure recognition is, in many ways, similar to the task of handwriting recognition. Hand-written character recognition is a prototypical application of machine learning, with the MNIST hand-written digit dataset serving as an archetype for assessing new classification algorithms.[7] Large variation in writing styles, poor image quality, lack of labelled data and cursive letters make hand-written text recognition a challenging task.[8-11] Hand-writing recognition falls into two camps: *online* recognition, in which a user writes text on a tablet or phone and it is recognized in real-time, and *offline* recognition, which refers to static images of hand-written text. Offline recognition poses considerably more challenges than online recognition due largely to the latter's ability to use time dependent strokes in combination with the final image to distinguish between characters.[12] In this work, we focus on offline hand-drawn hydrocarbon structure recognition, extending the potential use cases to digitization of lab notebooks.

Automatic extraction of a molecule from an image of its 2D chemical structure to a machine-readable format, termed *optical chemical structure recognitio*n, first emerged in the 1990s.[13-18] These systems were developed with the intent of mining ChemDraw type diagrams in the chemical literature to utilize the wealth of largely untapped chemical information that lies within publications. Over the following decades, more complex systems were developed, often based on the principles of their predecessors.[18-29] OSRA was the first chemical structure recognition open-source software, allowing new programs to be developed by direct extension. The majority of optical chemical structure recognition packages, including Kekulé,[15] IBM's OROCS,[16] CLiDE[17] and CLiDEPro,[22] ChemOCR,[21] OSRA,[23] ChemReader,[24] MolRec,[26] ChemEx,[27] MLOCSR,[28] and ChemSchematicResolver[29] rely on a rule-based workflow rather

than a data-driven approach. These systems achieve various degrees of accuracy, with the recently developed ChemSchematicResolver reaching 83-100% precision on a range of datasets.

Rule-based systems often involve complex, interdependent workflows, which can make them challenging to revise and extend. Therefore, several optical chemical structure recognition packages have been recently proposed based on data-driven, deep learning techniques.[30-32] Notably, Staker et al[30] employed an end-to-end image to molecule neural network, and ChemGrapher[31] used as series of deep neural networks to extract molecules from the chemical literature. Since these models are built directly from the provided data, they can be adapted or extended by presenting the neural network with different or additional training data, without the need for algorithm modification. Therefore, if there is an available dataset, data-driven systems offer a promising alternative to rule-based systems for this task.

The optical chemical structure recognition systems mentioned thus far focus on recognition of computer generated, ChemDraw-type structures. A handful of promising online hand-drawn chemical structure recognition programs have recently been developed,[33-34] however they currently remain limited in accuracy and generalizability. Our goal of extracting molecules from photographs of hand-drawn chemical structures is a further challenge. We believe that machine learning models are well-suited to address the noisiness of hand-drawn structures by augmenting and degrading the training data.

In this article, we begin by discussing our chosen deep learning approach for hand-drawn chemical structure recognition and demonstrate proof-of-concept on ChemDraw type images of molecules produced with the RDKit. Next, we describe the generation of two datasets: a small set of real-world photographs of hand-drawn hydrocarbon structures and a large synthetic dataset. We perform a series of experiments with these datasets, aiming to optimize the recognition accuracy on out-of-sample real-world hand-drawn hydrocarbons. We end by forming an ensemble model consisting of a committee of NNs trained in the experiments, which leads to a significant boost in recognition accuracy and introduces a confidence value for the predicted molecule. The work serves as a prototypical case study for approaching a chemical problem with machine learning methods, focusing on the explanation of deep learning, synthetic data generation, and ensemble learning techniques.

## Theory

Using molecules as the input or output of NNs require them to be represented in a machine-readable format. There are many ways to represent chemical structures, however each approach suffers from shortcomings. As a result, formulating new molecular representations has received significant attention in recent years.[35-36] The simplified molecular-input line-entry system (SMILES) format represents the molecular graph as a string.[37] Unfortunately, small changes in SMILES strings can lead to large changes in chemical structure (and often invalid molecules), and small changes in structure can lead to large changes in the string. As a result, representing molecules as graphs has become increasingly popular, with graph convolutional neural networks often being employed for encoding and decoding.[38-39] Other molecular representations include Cartesian coordinates and SELFIES.[36, 40-41]

In this work, we represent molecules as SMILES strings in order to leverage recent advances in natural language processing (NLP).[42] We employ neural image captioning, in which an image is input into a NN and a caption for the image is produced.[43-44] Here, an image of a hydrocarbon molecule is input and the predicted SMILES string is output, as shown in Figure 1. The NN architecture consists of a convolutional neural network (CNN)[45-46] encoder and a long short term memory (LSTM)[47] decoder. CNNs contain 'convolutional layers' that apply a convolutional filter over the image and pass the result to the next hidden layer; they are used primarily for encoding images since they conserve the spatial relationship of the pixels. LSTMs are a type of stable recurrent neural network (RNN) that make use of "gates" to learn long term dependencies, popularly used for language applications. This is a useful feature in the case of decoding SMILES strings since there are often relations between characters at the start and end of the string, such as closing of a parentheses pair to indicate the end of a branching group. Our image-to-SMILES approach is inspired by the work of Deng et al., which trained a NN to convert images of mathematical formulas to LaTeX code.[48] A similar approach was also used by Staker et al. to recognize SMILES strings from ChemDraw type images in the chemical literature.[30]

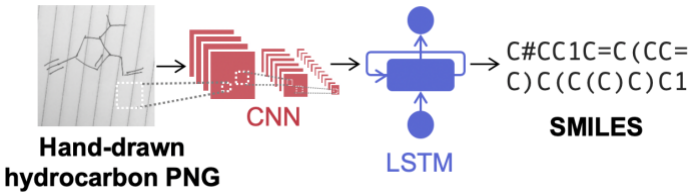


**Figure 1.** Image-to-SMILES neural network used for hand-drawn hydrocarbon recognition based on neural image captioning. The network consists of a convolutional neural network (CNN) encoder and long short-term memory (LSTM) decoder network.

Chemical structure recognition is a *supervised learning* problem: each input is associated with an output label. During the training process, a defined loss function, which depends on the error between the predicted NN label and the reference label, is minimized. Our image-to-SMILES network is an example of an encoder-decoder network: the input is encoded to create a compressed representation of the data, which is subsequently decoded to the predicted output. The key idea behind these encoder-decoder workflows is the ability to learn a mapping between two different representations of the same data by compressing it to its key "features" through the central bottleneck called the *latent space*. One of the properties of encoder-decoder frameworks that has made them so successful for supervised learning is the ability for the same network architecture to be used for many different applications by simply providing data specific to that application. For example, a CNN-LSTM network can be applied to chemical structure recognition, image caption generation,[44] mathematical formula recognition,[48] optical character recognition (OCR) in natural scenes[49] and hand-writing recognition[11] to name a few. Moreover, since the encoder and decoder networks are swappable components, they generalize well beyond the CNN-LSTM applications: machine translation can be achieved by simply swapping the CNN with another LSTM to form a sequence-to-sequence model for instance. Autoencoders are a special case of encoder-decoder networks in which the target output space is equal to the input space, used as a way of performing dimensionality reduction.

In our image-to-SMILES network, the LSTM decoder uses an *attention* mechanism to improve the accuracy of the output text sequence.[44, 50-51] The attention mechanism learns a probability mask over the image by calculating a "context vector" which acts as a dynamic pointer to relevant areas of the image during decoding. This reduces the loss of higher-level image features at the encoder bottleneck. For example, a high attention score for pixels showing two parallel lines in the chemical structure might prompt "=" to be output from the LSTM.

In addition to attention, beam search was also used in the decoding layers. Beam search keeps track of the top *k* most probable NN predictions. During training, RNNs output predicted characters of the SMILES string and pass them back into the network, which outputs the next predicted character and so on. Applying beam search to an RNN allows the network to keep track of the strings with the *k* highest cumulative probability at each decoding step, while the other predictions are pruned. The final output of the NN is a list of length *k* with the highest ranked predictions. A *greedy* decoder would have $k = 1$, meaning that only the highest probability characters are saved.

## Methods

### *Neural network architecture*

For the NN training we applied an adapted sequence-to-sequence (seq2seq) model used originally for mathematical equation recognition.[52] The CNN encoder architecture outlined in Table S1 was implemented. An LSTM with 512 units and embedding dimension of size 80 was used for decoding, with beam search ($k = 5$) and attention mechanism intermediary vector dimension of 512. We used the Adam optimizer[53] and a batch size of 20 for training. Network weights were saved based on the validation set's perplexity, *p*, calculated as

$$p = -\exp\left(\frac{H_{chars}}{n_{chars}}\right)$$

where $H_{chars}$ is the sum of the cross-entropy loss for the characters in the validation set, and $n_{chars}$ is the number of characters in the validation set. A learning rate of $1\text{x}10^{-4}$ was used for all training runs and the model was implemented in Tensorflow.[54] We define the NN *accuracy* as the proportion of molecules predicted exactly correctly, i.e., the predicted SMILES matches the target SMILES character-by-character.

### *Datasets*

We extracted a dataset of 500,000 SMILES strings with a ring size of less than eight carbon atoms from the GDB-13 and GDB-11 databases.[55-57] The vocabulary was restricted to "Cc=#()1", where = and # indicate double and triple bonds, respectively, parentheses indicate the start and end of a branching group, lower case letters represent aromaticity, and numbers are found at the start and end of rings. To remove ambiguous skeletal structures from our dataset that confuse the NN during training, we only include the number '1' meaning that molecules with

multiple conjoined rings are not considered. The SMILES labels were canonicalized using RDKit to give a single target output. After canonicalization, molecules outside of the vocabulary were removed, resulting in a ~10% reduction in size for all datasets used in the experiments presented. RDKit was used to generate images of molecules from the SMILES dataset, by first generating SVG files and then converting to PNG format. The result is a labelled dataset of image and SMILES pairs; representative examples are shown in the Figure 2 inset. We used this clean RDKit dataset to perform proof-of-concept for the image-to-SMILES network. A synthetic dataset based on RDKit images designed to mimic hand-drawn data was curated for the purpose of this study. We discuss the auxiliary data generation workflow, and experiments performed on this dataset in the coming sections.

The computer-generated datasets were first split into a 90% training/validation set, and a 10% test set. The test set serves as out-of-sample data used to evaluate the accuracy of the network after finishing the training process. The training/validation set, used during training, was then split further into a training set (90%) and a validation set (10%). The real-world photographs of hand-drawn hydrocarbons consisted of a total of 613 images. We set aside a 200-image test set, with the remaining 413 images being either used entirely as a validation set or split into validation (200 images) and training (213 images) datasets, depending on the experiment. All images were resized to 256 x 256 pixels and converted to PNG format using OpenCV.[58]

## Results and discussion

### *Synthetic data generation*

To test the suitability of our image-to-SMILES network for hand-drawn molecule recognition, we begin by training with clean images of hydrocarbon skeletal structures generated with RDKit and their respective SMILES labels (Figure 2). In order to determine the dataset size required to achieve a given recognition accuracy, the NN was trained with datasets of size $10^4$, $5x10^4$, $10^5$, $2x10^5$ and $5x10^5$ images (split between training, validation and test sets as described in the methods section). The results of the proof-of-concept training are shown in Figure 2, illustrating the increasing NN recognition accuracy with dataset size. A dataset of $5x10^4$ labelled RDKit images achieves an out-of-sample (test set) accuracy of over 90%, and a maximum

accuracy of 98% is achieved with a dataset of $5x10^5$ images. This demonstrates that the chosen NN architecture is capable of learning SMILES strings from images of hydrocarbons.

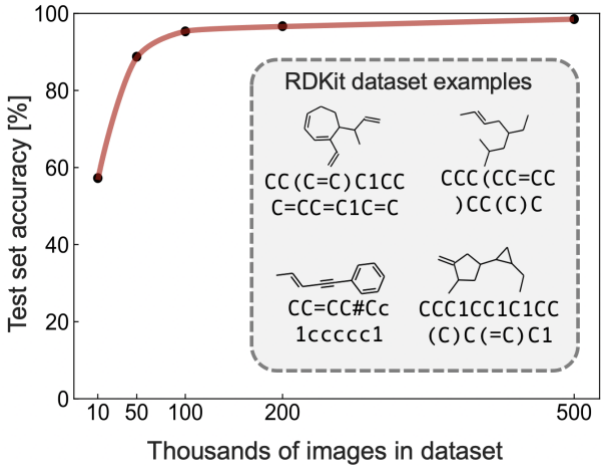


**Figure 2.** Out-of-sample accuracy of the image-to-SMILES network trained with an increasing number of clean RDKit hydrocarbon structures and their corresponding SMILES label. Representative examples of labelled RDKit training images and SMILES are shown in the inset.

Although the results from training with RDKit images suggest that a dataset of $5x10^4$ images obtains 90% out-of-sample accuracy, in reality a much greater number of *hand-drawn* hydrocarbon molecules are likely needed to achieve this same accuracy. As with handwritten text recognition, variation in drawing style, backgrounds and image quality provide significant challenges. There is noise associated with (i) the chemical structure, such as varying line widths, lengths, angles and distortion, (ii) the background, such as different textures, lighting, colors and surrounding text, and (iii) the photograph, such as blurring, pixel count and image format (Figure 3). A further challenge of chemical structure recognition is the ability for a molecule to be drawn in any orientation, in contrast to text recognition of languages written in one direction, e.g. left-to-right.

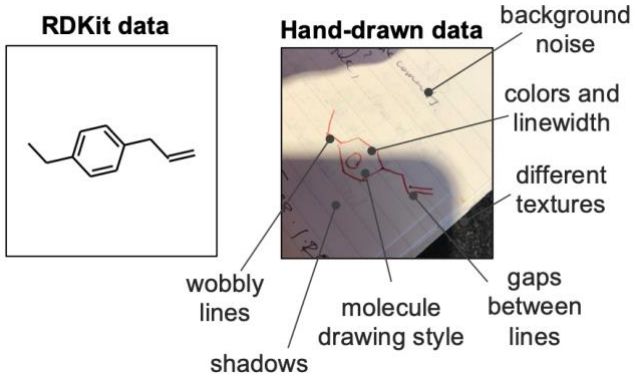


**Figure 3.** Comparison between (a) a computer-generated (RDkit) image of a hydrocarbon structure and (b) a photographed hand-drawn hydrocarbon structure. The differences between the two images are highlighted, demonstrating the increased complexity of hand-drawn structure recognition.

Since end-to-end NNs learn a model solely from the data presented during training, access to high-quality data is imperative to achieve an accurate model. Unfortunately, a labelled dataset of real-world hand-drawn molecules does not exist and cannot be easily generated. Therefore, unlike in the case of RDKit images, it is not possible to achieve high recognition accuracy by simply training with hundreds of thousands of hand-drawn structures. Lack of training data is a common hurdle when attempting to apply end-to-end deep learning models to real-world problems. This is especially true in fields where producing datapoints is time and energy intensive, such as the chemical domain. In cases such as these, generating synthetic data can prove more efficient than spending excessive time and resources collecting large amounts of real-world data.

We developed a data collection web app to source a small dataset of hand-drawn chemical structures. In order to capture the large noise in drawing style, photograph quality and background types that are prevalent in real-world data, we collected data from many different drawers by promoting the app to a range of groups in the Stanford University Chemistry Department. This aimed to reduce the risk of the network learning to recognize only a single user's drawing style. Over 100 unique users of the app generated over 5800 photographs of hand-drawn chemical structures, 613 of which were hydrocarbons. Details of the data collection app are shown in Figure S1 and the collected dataset will be released with this paper. Based on our earlier RDKit image results (Figure 2), ~ 600 images is several orders of magnitude less data than necessary to train to any reasonable recognition accuracy. As a result, in addition to

sourcing real-world data, we also developed a workflow to generate a large synthetic dataset to be used in conjunction with the limited real-world dataset for training.

An ideal synthetic dataset is exactly equivalent to the target data, but can be readily generated on large scales (unlike the target data). The desired datatype (of which there is insufficient data for training) could therefore be substituted with synthetic data during training and the weights would be directly transferable to the target data. To discuss how to generate such an auxiliary dataset, we consider a subspace that spans from the desired datatype to a similar, readily scalable datatype. In our case, this is the subspace between photographs of hand-drawn molecules and RDKit images. The aim is to find a mapping that moves both datatypes to the same point in the subspace such that they are indistinguishable. Figure 4 depicts such a subspace, highlighting possible convergence routes. Perhaps the most obvious pathway transforms raw RDKit data (bottom right) into images that resemble raw hand-drawn data (top left) as closely possible (or visa-versa); this might involve adding in backgrounds, distorting the lines and blurring the image. Indeed, it is also possible to modify *both* datatypes such that they reach a common point in the subspace that lies away from both the original data points – so long as the two datatypes converge to the same point, they are equivalent. For example, applying *edge detection* (or *background removal*) to both the hand-drawn and computer-generated data would result in movement away from their respective raw datatypes, but closer to one another. In this instance, a model would be trained with an edge-detected synthetic dataset, and later applied to hand-drawn hydrocarbon molecule images that have been pre-processed with edge detection. Mapping two datatypes to a common point in a subspace is commonly used in deep learning applications since there is often a limited amount of the exact data needed, but a similar readily accessible datatype that can form the basis of a synthetic dataset.[11, 59-60]

It is important to note that, although the desired and synthetic datapoints should converge, the data must also maintain enough structure to allow the SMILES string to be extracted from the image. In other words, the information content, i.e., the important features, must be preserved. For example, consider the extreme case of setting all the pixels in the image to black for both datatypes: the data would reside at the same point in the subspace, however the NN would not be able to learn the mapping from image to SMILES. A one-to-one mapping between the two datatypes and the output label must exist, i.e., one image should only correspond to exactly one molecule.

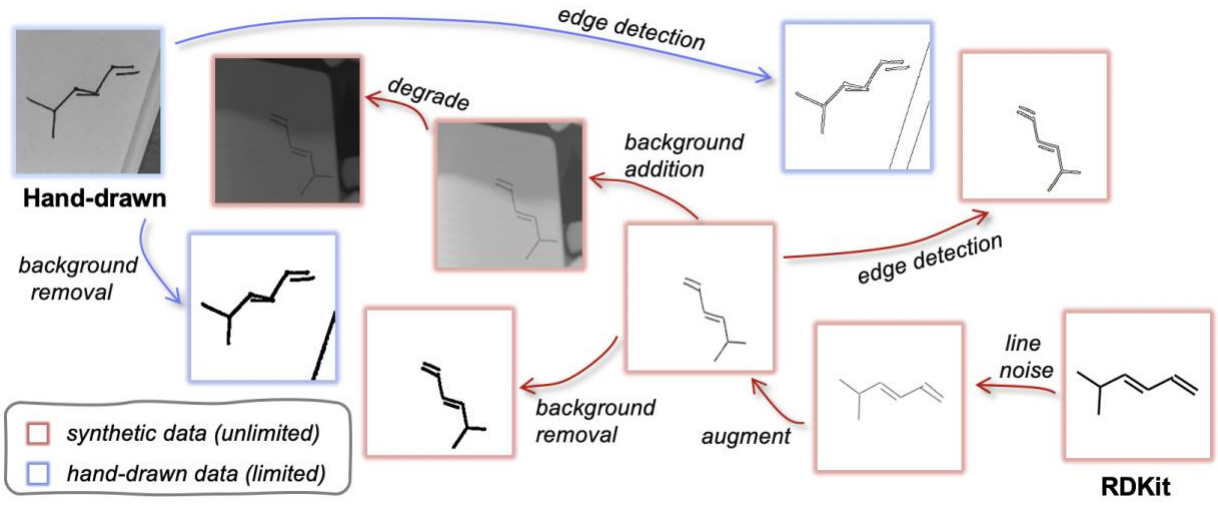


**Figure 4.** Data subspace that spans from target data (photographs of hand-drawn hydrocarbon chemical structures, top left) to scalable data (raw RDKit images, bottom right). Paths to reach similar points in the subspace for the target data and synthetic data are demonstrated. Blue outline: hand-drawn images, red outline: computer-generated images.

We explored auxiliary datasets based on background removal and edge detection algorithms, however these image processing techniques were often found to be brittle when applied to real-world hand-drawn data. For example, dark shadows, lined paper and thin pencils made it hard to clearly identify the molecule after applying such algorithms (Figure S2). To ensure the recognition software is robust to a wide range of potential images, for the remainder of this study we focus on generating a synthetic dataset that resembles hand-drawn molecules as closely as possible. Figure 5a outlines the synthetic data generation workflow developed to transform RDKit images into synthetic photographs of a hand-drawn hydrocarbon structure. First, we introduce randomness to bond angles, lengths and widths via modification of the RDKit source code (RDKit'). The image is then passed through the *augmentation pipeline* that applies a series of random image transformations according to a defined probability (RDKit'-aug). The augmented molecule image is then combined with a randomly augmented background image by weighted addition with OpenCV (RDKit'-aug-bkg). Finally, the image is passed through a degradation pipeline to form the final synthetic data (RDKit'-aug-bkg-deg). The molecule augmentation, background augmentation and image degradation workflows are outlined in Figure 5b, with all the transformations applied in these pipelines detailed in Table S2. Generating

a synthetic datapoint from a SMILES string takes ~1s, hence, over 85,00 labelled images of hydrocarbons can be produced in 24 hours of compute time. For comparison, it takes ~1 minute for a human to draw, photograph, and label a hydrocarbon chemical structure, meaning that ~2 months of continuous human effort would be needed to achieve a dataset of this size.

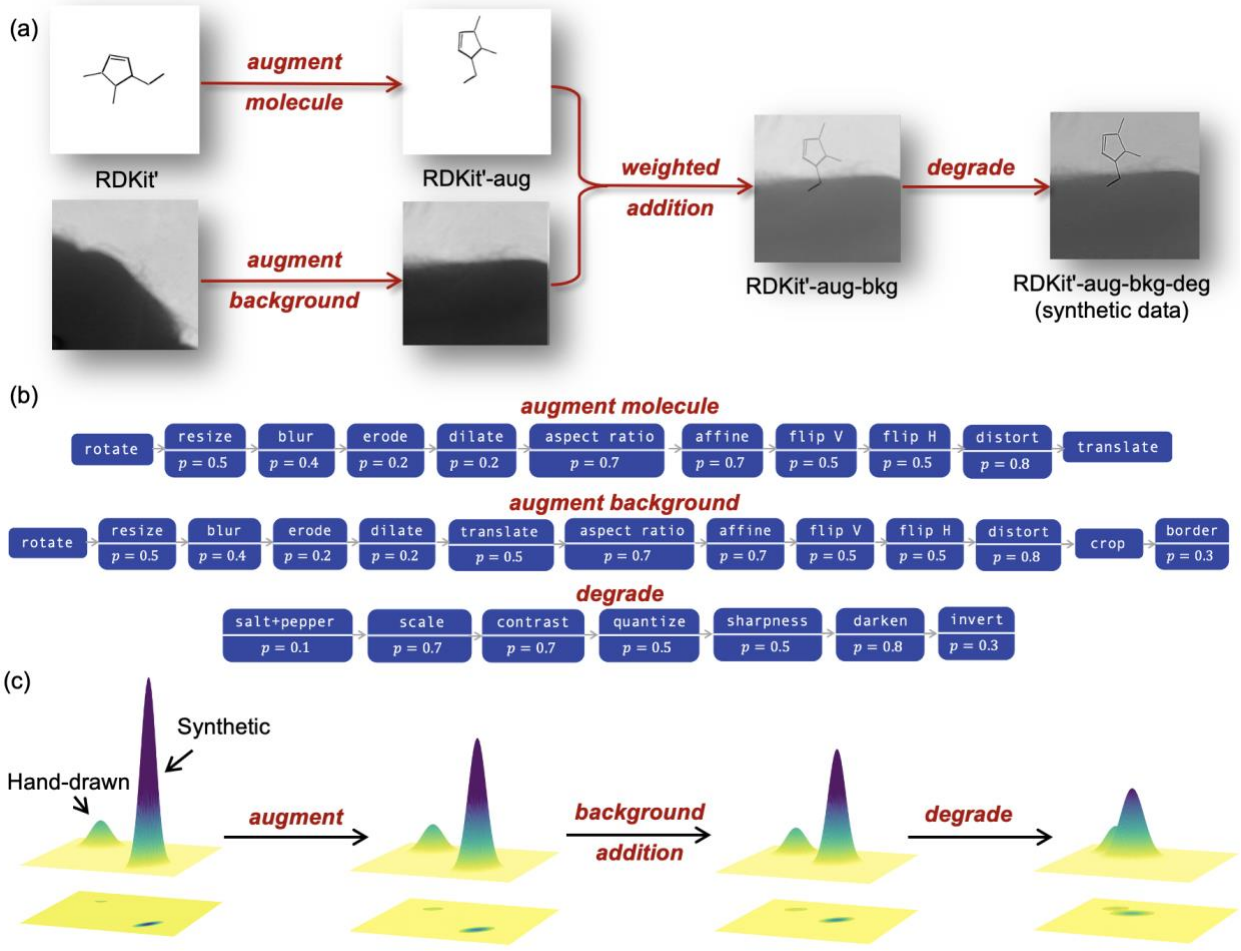


**Figure 5.** (a) The synthetic data generation workflow with the datatype's assigned name for each stage of the pipeline. (b) The augment molecule, augment background and degradation pipelines used for the synthetic data generation. Each box corresponds to a function that is applied according to probability, *p*. A complete list of the image transforms associated with each function is given in the Supplemental Information. (c) Schematic depiction of how the steps in the synthetic data workflow move the synthetic data distribution towards the hand-drawn data distribution by representing the datasets as two-dimensional gaussians (not to scale).

The background images are randomly selected from a dataset of 1052 photographs (Figure S3). This backgrounds dataset was collected relatively easily as it did not require labelling. By

adding the photographed backgrounds to a known molecule, a labelled synthetic dataset with realistic background textures and photograph features is produced. Since it is common for the act of labelling data to be the most time intensive step of dataset generation, sourcing a large dataset of an unlabelled component of the data and combining it with a synthetic labelled component can be an inexpensive way of generating synthetic data with realistic features.

The molecule and background image augmentation pipelines (Figure 5b) introduce noise into the data through rotations, translations, distortion and other image transformations. This acts as a form of regularization during training to reduce overfitting (where the NN reaches high accuracies during training but much lower accuracies on out-of-sample data). The importance of broadening the data distribution can be exemplified with background augmentation: without augmenting backgrounds the NN may become overly familiar with the structure of the background images used during training and learn to remove them from the image. The result is bad generalization when presented with images that have different backgrounds to those seen during training. The augmentations are deliberately more aggressive than what would be found in real-world images to span the maximum dataset subspace, i.e., make the distribution as wide as possible.

In addition to augmentation, we also randomly degrade the data to further increase the regularization. This accounts for features like variation in image quality and type. The degradation pipeline was adapted from work by Ingle et al.,[11] which leveraged a large dataset of online data for offline hand-written text recognition by applying aggressive degradation. Through a well-constructed degradation workflow, they were able to achieve a large increase in accuracy, particularly for cases where a small number of real-world images were available. We will later show the effect that data augmentation and degradation have on our recognition accuracy.

As described previously, the stages of the synthetic data generation pipeline are designed to map the synthetic distribution onto the distribution of real-world hand-drawn chemical structures. A simplified schematic of how each step effects the data distribution is shown in Figure 5c. The datasets are represented as two-dimensional gaussians, with their amplitude proportional to the quantity of data and their width proportional to the data variation within the distribution. As the data proceeds through the augmentation, background addition and degradation steps, the synthetic distribution approaches the hand-drawn data distribution in the

subspace, as was previously explored in Figure 4. We also show widening of the distribution as these steps are applied in order to span as wide a range of data as possible to minimize overfitting.

Representative examples of the synthetic and real hand-drawn datasets are compared in Figure 6. By eye, the synthetic images strongly resemble the hand-drawn data. However, since NNs read the images as an array of pixel values, an important comparison metric is the frequency of the pixel values found in the images. We do this by comparing histograms of pixel intensity, which ranges from 0 (black) to white (255), for the synthetic and hand-drawn data. It can be seen that the synthetic data often has less of a smooth, continuous pixel count and less texture than the real-life data. Also, the frequency of pixel intensities is generally higher for the synthetic data in comparison to the hand-drawn data. These differences are due to the heavy augmentations of the backgrounds in the synthetic data pipeline (e.g., cropping and adding borders) which results in reduced image texture. This discrepancy could be reduced by increasing the size of the background dataset such that less aggressive augmentations would be required.

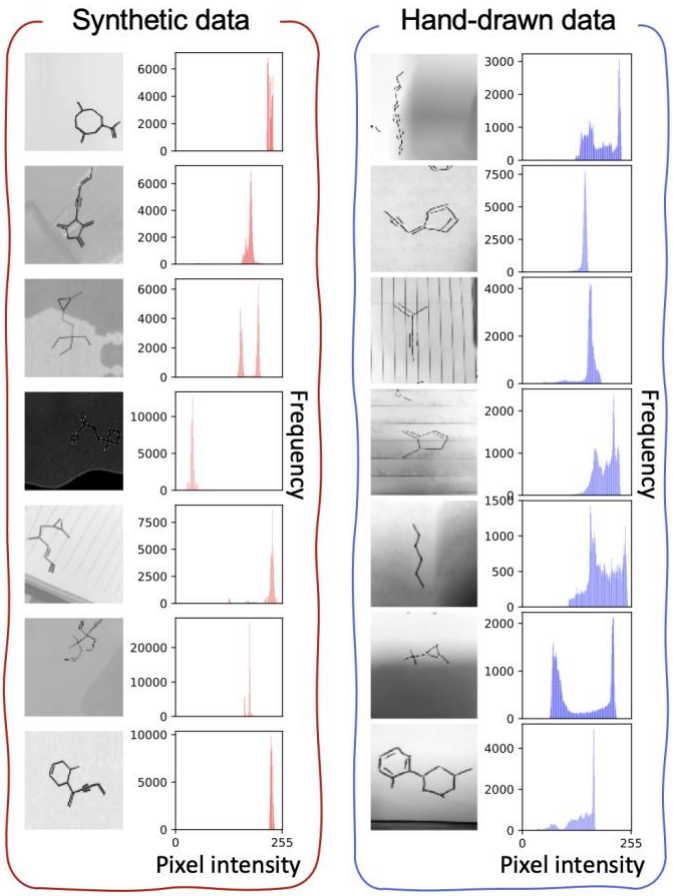


**Figure 6.** Comparison of representative images taken from synthetic dataset (left) and real-life hand-drawn dataset (right) and their pixel count histogram calculated by flattening the greyscale images and calculating the frequency of pixel intensities (values from 0 to 255 where 0 is black and 225 is white).

Analysis of the pixel counts highlights the dangers of over-augmentation. A compromise must be reached between augmenting enough to prevent overfitting, but not so much that the data no longer resembles the target. An example of an overly-augmented background image can be found in Figure S5, which was generated with heavily cropped backgrounds. Since we have access to only a limited number of background images, we choose to augment our synthetic data relatively aggressively. However, we limit overly excessive cropping and resizing so not to remove completely the continuous texture of the image. Heavy augmentation can also lead to uninterpretable data, for example, molecules may be distorted such that bonds cannot be distinguished (Figure S5). This can confuse the training process and result in an increased error rate.

*Neural network experiments*

In the following section we lay out a series of experiments designed to understand how our real-world and synthetic datasets can be best utilized to achieve the highest out-of-sample hand-drawn hydrocarbon recognition accuracy. First, only synthetic data is used during training; we investigate how the synthetic data generation pipeline, and training set size impact the NN accuracy. Once we have an understanding of the synthetic data results, hand-drawn data is introduced into first the validation set, and then the training set. The results of fine-tuning are compared to training from scratch. We end by forming an ensemble model from the trained NNs, which allows us to assign a confidence value to the prediction, as well as improve the recognition accuracy.

In order to examine how each stage of the synthetic data generation workflow (Figure 5a) affects training, we train our CNN-LSTM network with data from each stage of the pipeline: modified RDKit images (RDKit'), augmented RDKit images (RDKit'-aug), augmented RDKit images with background addition (RDKit'-aug-bkg), and augmented RDKit images with background addition and degradation (RDKit'-aug-bkg-deg). Datasets of 200 000 images from each of the four steps in the workflow were split into train, validation and test sets as detailed in the Methods section. The results of the training are presented in Figure 7a. As the steps proceed through the synthetic data generation pipeline, the non-uniformity of the data increases, making it more complex, and hence more challenging for the NN to learn. As a result, slower optimization and a reduction in final accuracy is observed (Figure S6). The image-to-SMILES network is tested on the same datatype used for training (e.g. if the network was trained with RDKit'-aug data, it would also be tested on RDKit'-aug data) as well as our real-life hand-drawn dataset. As discussed previously, the synthetic data pipeline was developed in order to match the hand-drawn data as closely as possible. The test set accuracy of the hand-drawn data is seen to increase from 8% to 47% as we proceed through the steps in the data generation pipeline, illustrating that each stage performs the desired effect of bringing the computer-generated data and the hand-drawn data distributions closer together.

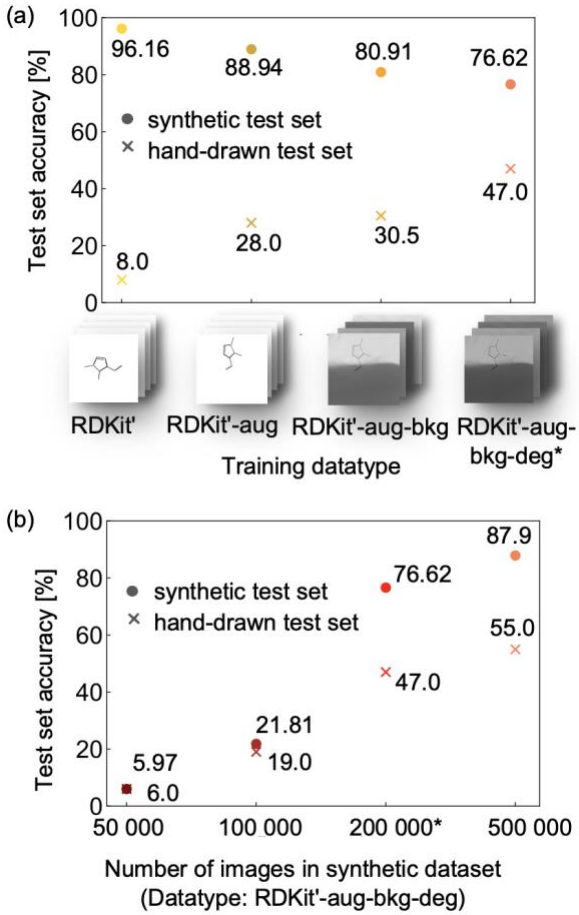


**Figure 7.** Training experiments with synthetic data showing the test set accuracy of the datatype used for training (x) and hand-drawn data (o). (a) Results of training with 200 000 images from each stage of the synthetic data generation pipeline: modified RDKit images (RDKit'), augmented RDKit images (RDKit'-aug), augmented RDKit images with background addition (RDKit'-aug-bkg), and augmented RDKit images with background addition and degradation (RDKit'-aug-bkg-deg). (b) Results of training with different sized training sets of the final synthetic data (RDKit'-aug-bkg-deg). Equivalent training runs in the two sets of experiments are indicated (*).

From Figure 7a, it can be seen that augmenting the images increases the hand-drawn hydrocarbon recognition accuracy by 20 percentage points. Moreover, degrading the images increases the accuracy from ~30% to nearly 50%. The jump in accuracy when adding in backgrounds is minimal in comparison to the addition of augmentation and degradation. This is a surprising result, since observation by human eye suggests that background addition would move the synthetic data significantly closer to the hand-drawn hydrocarbon target data, in comparison to introducing image degradation. This serves as a powerful demonstration of the importance of

evaluating the success of the auxiliary data techniques though training experiments rather than by eye and the effectiveness of broadening the data distribution via augmentation and degradation.

Next, we investigate how the size of our synthetic dataset (RDKit'-aug-bkg-deg) impacts the training and test set accuracies. The network was trained with datasets of size 50 000, 100 000, 200 000 and 500 000 images (split between training, validation and test sets according to the Methods section). As the number of images in the synthetic dataset increases, the out-of-sample recognition accuracy on the synthetic data grows from 0% to nearly 90% (Figure 7b). A particularly large leap (~20% to ~75% Acc.) is seen from the 100 000 to 200 000 image datasets and a smaller jump between 200 000 and 500 000 images is observed as the recognition accuracy begins to plateau. It can also be seen that the difference between the accuracy of the synthetic and hand-drawn test sets increases with dataset size, demonstrating how the network begins to overfit to the synthetic data. Remarkably, the NN trained with 500 000 images achieves an accuracy of over 50% on real-world hand-drawn data, despite not having been exposed to hand-drawn data at any point during the learning process. This result suggests that the auxiliary data bears significant resemblance to the target datatype, and hence assigns some confidence that the workflow developed in the previous section behaves as desired. For reference, training with 500 000 raw RDKit images results in a real-life hand-drawn hydrocarbon recognition accuracy of 0%.

Now that we have examined how the stages of the synthetic data generation pipeline and dataset size effect the network's recognition accuracy, we can begin to incorporate photographs of real-world hand-drawn data into the training process i.e., the training and/or validation sets. The aim is to explore how best to utilize a limited target dataset and a large synthetic dataset to achieve the highest accuracy on out-of-sample target data. Since the model is constructed based on only the data used for training, if the training data more closely matches the desired testing data the model will perform better. After each epoch, the NN weights are tested on the validation set to track the model's accuracy as the training proceeds. Weights that achieve the best results are saved according to the network's perplexity score, a measure of the uncertainty of the prediction. Although the validation set is not directly used for optimizing the weights, it can be thought of as a "target" that the NN is aiming for. Therefore, this target should be equivalent to the desired use case such that the maximum out-of-sample accuracy for the desired application is

reached. As a result, we expect that adding hand-drawn structures to the training and validation sets increases the hand-drawn molecule recognition accuracy.

We examine the effect of replacing the synthetic validation set with a 213-image hand-drawn validation set. The size of the synthetic training set was varied from 50 000 to 500 000. As with the previous results, the hydrocarbon recognition accuracy increases with the size of the synthetic dataset (Figure 8a). Comparison to Figure 7b, which used a synthetic validation set instead of the hand-drawn data, shows very similar accuracies. Therefore, for this task, employing hand-drawn data as a validation set does not result in any significant increase of the hand-drawn hydrocarbon structure test set accuracy. A notable difference, however, lies in the training mechanism: when using a synthetic train and validation set, as in the previous experiments, the NN overfit to the synthetic data – the training accuracies were high compared to the lower test set accuracies. In contrast, training with a synthetic training set and hand-drawn validation set does not overfit – the training and test set accuracies are comparable – however the limitations of the synthetic data prevent it from training to higher accuracies. This can be seen by comparing the training accuracies in Figure S6b to the test set accuracies in Figure S7a. Despite the contrasting mechanisms, the trained networks produce similarly successful test set outcomes.

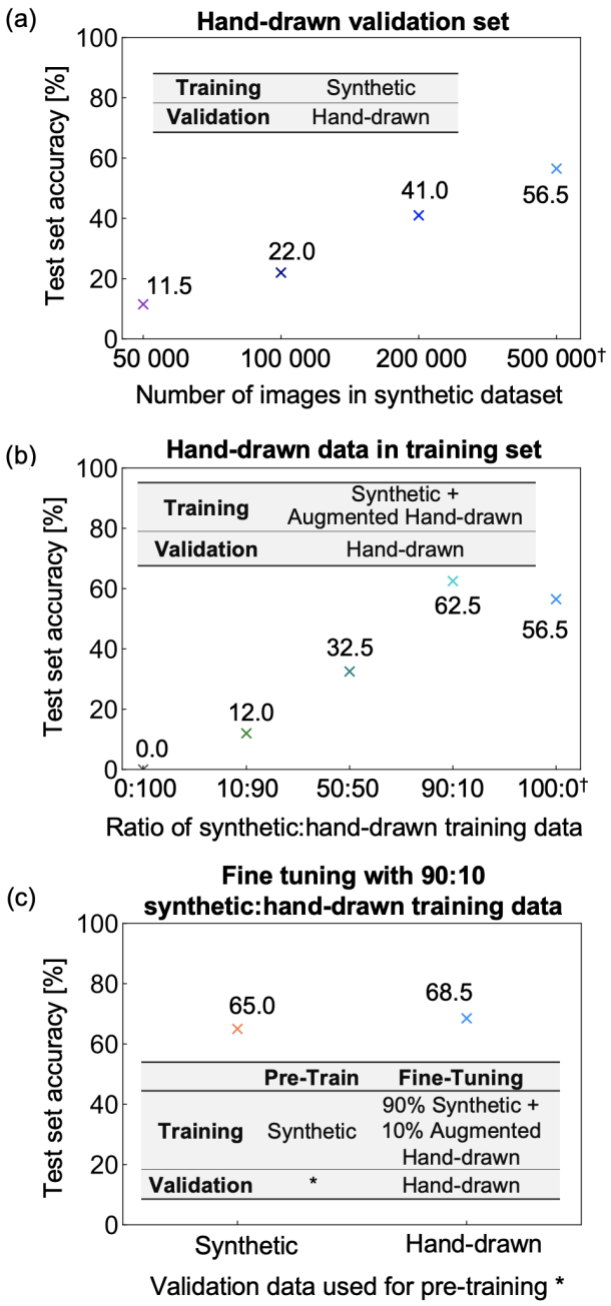


**Figure 8.** Recognition accuracy of the hand-drawn hydrocarbon test set after incorporating hand-drawn data into the training process. (a) Results of training with increasing sized synthetic training sets and a hand-drawn hydrocarbon validation set. (b) Results of training with varying ratios of augmented and degraded hand-drawn hydrocarbon to synthetic data training sets (500 000 image total) and a hand-drawn hydrocarbons validation. (c) The effect of fine tuning is investigated by restarting the weights from training with a 500 000-image synthetic dataset used for both training and validation, and a 500 000-image synthetic dataset used for training with hand-drawn validation set. The weights are restarted with a training set consisting of 90% synthetic data and 10% augmented and degraded hand-drawn data, and a validation set of hand-drawn hydrocarbons. Equivalent training runs in the experiments are indicated (†).

We now incorporate hand-drawn data into the training set so that it can directly impact the weight optimization during training, allowing the NN to *learn* from the target data, rather than only determine if the weights should be saved. The number of remaining images of hand-drawn hydrocarbon structures in our dataset after the removal of the test set is 413, which must be distributed between the training set and validation set. We assign 213 images to the training set and 200 images to the validation set. A dataset of 500 000 images is chosen since it reached the highest accuracies in our synthetic data experiments.

We trained the image-to-SMILES network with varying ratios of augmented and degraded real-world hand-drawn to synthetic data, and tested the weights on the hand-drawn test set data (as used in earlier experiments). Due to the very limited hand-drawn hydrocarbon data, we augmented and degraded the images to produce the number needed in the training set to satisfy each given ratio. For example, to generate a training set of 50% hand-drawn and 50% synthetic images (250 000 images each), each hand-drawn image was augmented ~1173 times using the *augment molecule* pipeline (Figure 5b, excluding the final translation step). Although this introduces a large number of repeated SMILES and similar images, the small amount of hand-drawn data makes this necessary to ensure that the information is not overridden by the large amount of synthetic data. Once the molecules have been augmented and degraded, the synthetic and hand-drawn data are randomly shuffled together for training.

We investigate ratios of 0:100, 10:90, 50:50, 90:10 and 100:0 synthetic:hand-drawn data. From Figure 8b, it can be seen that using entirely hand-drawn data results in an out-of-sample accuracy of 0% due to the network overfitting to the very narrow distribution of hand-drawn training data. Adding synthetic data allows the neural network to be exposed to many more molecules and image types, and hence leads to a rapid increase in test set accuracy up to 90:10 synthetic:hand-drawn data. Removing the final 10% of hand-drawn hydrocarbon molecules from the training set (equivalent to the 500 000 image training run presented in Figure 8a), however, leads to a decrease in the hydrocarbon recognition accuracy from 62% to 56%. Therefore, the results suggest that two opposing effects are at play: (i) including target data in the training set allows the weights to be optimized for the target application and (ii) including only a narrow or sparse distribution of target data leads to overfitting. As a result, including a small portion of target data, specifically 10% hand-drawn molecules, yields the highest recognition accuracy.

In all the experiments discussed so far, the image-to-SMILES network has been trained from scratch, i.e., the weights are randomly initialized. When applying deep learning to tasks with limited available data, training the network with a large dataset before restarting the weights with a similar dataset has been shown to increase NN accuracy.[61] This approach is termed *fine-tuning* due to the NN weights being tuned from a related task to better suit the desired datatype. Fine-tuning is a similar to *transfer learning*, which freezes a portion of NN layers weights during re-training.

We apply fine-tuning to our problem by first training with synthetic training data and then restarting the NN weights with training data that includes real-life images of hand-drawn hydrocarbon structures. We fine-tune two trained NNs, both of which use 500 000 image synthetic training datasets but that differ in their validation data: the first uses a synthetic validation set (pre-training results shown in Figure 7b) and the second uses a hand-drawn validation set (pre-training results shown in Figure 8a). The two trained NNs are restarted with a training set made up of 90% synthetic data and 10% hand-drawn data – the optimal ratio according to Figure 8b. The results from the two fine-tuning runs (Figure 8c) show that pre-training with synthetic data before incorporating hand-drawn data into the training set improves the molecule recognition accuracy. The network reaches 68% accuracy after pre-training with a hand-drawn validation set, in comparison to the best NN trained from scratch which achieved an accuracy of 62%.

*Ensemble learning*

Instead of relying on a single model to predict a desired output, combining several models can result in improved performance. The process of combining models to form an ensemble model is called *ensemble learning*.[62] There are several ways in which ensemble models can operate, such as boosting, bagging and random forests.[63] Perhaps the simplest ensemble, however, is a committee of trained NNs, where each NN casts a single vote according to their prediction. The ensemble model's predictions can then be ordered from most to least votes and the prediction corresponding to the most votes, i.e., the mode, is output. The number of agreeing votes for a prediction can give insight into the confidence of the ensemble model. If all of the models predict the same output, there is a high probability the prediction is accurate. However, if

all the models disagree, there is high uncertainty in the prediction. We demonstrate these properties by forming a committee of trained NNs.

We build two ensemble models, one comprised of all the trained NNs presented in the previous section (All-models), and another comprised of only trained NNs that achieve >50% accuracy on the hand-drawn test set (Top-models). The out-of-sample hand-drawn hydrocarbon recognition accuracy for each of the two ensemble models is shown in Figure 9, comparing the three predictions that have the most votes with the reference SMILES label. The "Top-models" ensemble model achieves an accuracy of 77% on the hand-drawn test set for the top prediction and 86% if the top three predictions are considered. The "All-models" ensemble achieves only slightly lower recognition accuracies for the top three SMILES predictions. By forming a committee of NNs, we see a significant improvement in accuracy in comparison to the constituent NNs (the highest of which obtained 68% on out-of-sample hand-drawn data). Examples of correctly and incorrectly labelled hand-drawn hydrocarbon structures are provided in Figure S8.

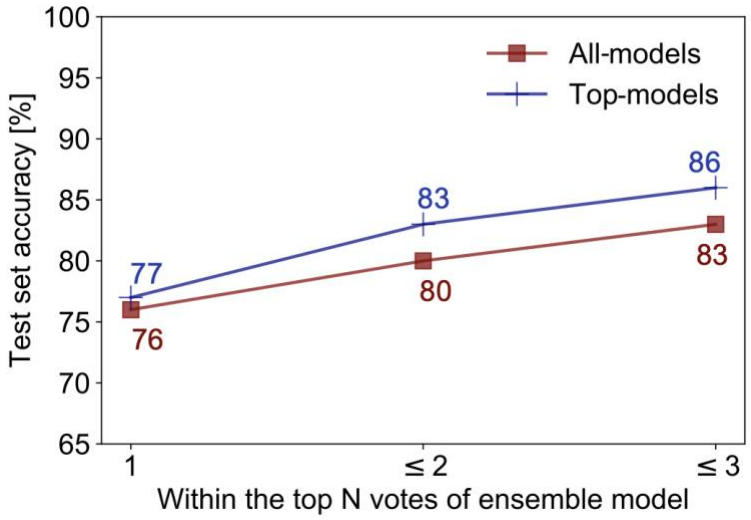


**Figure 9.** The out-of-sample hand-drawn hydrocarbon recognition accuracy of the top *N* predictions of two ensemble models made up of all trained NNs in previous experiments (All-models, red), and only trained NNs with over 50% recognition accuracy on out-of-sample images of hand-drawn hydrocarbon molecules (Top-Models, blue).

The agreement between the models that make up the committee offers insight into the certainty of the prediction. Figure 10a shows the increase of recognition accuracy as the number of votes for the top prediction, *V*, rises. Here, we assign the accuracy of the ensemble model

when there are $V$ agreeing votes to its confidence value. As expected, as $V$ increases, so does the accuracy (or confidence) of the model. When all the models disagree ($V = 1$) the model has low out-of-sample accuracy, equating to a low confidence value of the model. When many models agree, the prediction tends to have a higher accuracy. For the "Top-models" ensemble, all of the models agreeing ($V = 4$) translates to a confidence value of 98% in the predicted hydrocarbon.

In addition to knowing the confidence of the model's prediction, it is useful to be aware of how often the model achieves this confidence: if the model was 100% confident when all the votes agreed but this only occurred 1% of the time its use would be limited. We therefore investigate the portion of times that a confidence value occurs in the test set (Figure 10b). For the "Top-models" ensemble, the percentage of times that $V$ votes occurs, corresponding to a confidence value (Figure 9c), increases with the number of votes – there are few instances where all the NNs disagree ($V = 1$), and by far the most common occurrence is all NNs agreeing ($V = 4$). A similar trend is observed for the "All-models" committee.

The importance of knowing the uncertainty of a model's prediction should not be underestimated. In many cases, it is more important to achieve a lower accuracy but be able to predict when the model will fail, than to achieve a higher accuracy but have no insight into when it will fail. For example, in the case of autonomous vehicles, a model that is able to determine when it will fail and prompt a human to take over controls would be far safer than a model that failed less but was unable to forecast failure. In the case of hand-drawn molecule recognition, the software could, for example, prompt the user to take a second photograph if the uncertainty of the model was high. Of course, both the accuracy and confidence of the output should be optimized. Here, our ensemble model recognizes the correct molecule with near 100% confidence in over 50% of cases ($\geq 97\%$ in 55% of cases). This is a promising result for applying this technology to real-world applications.

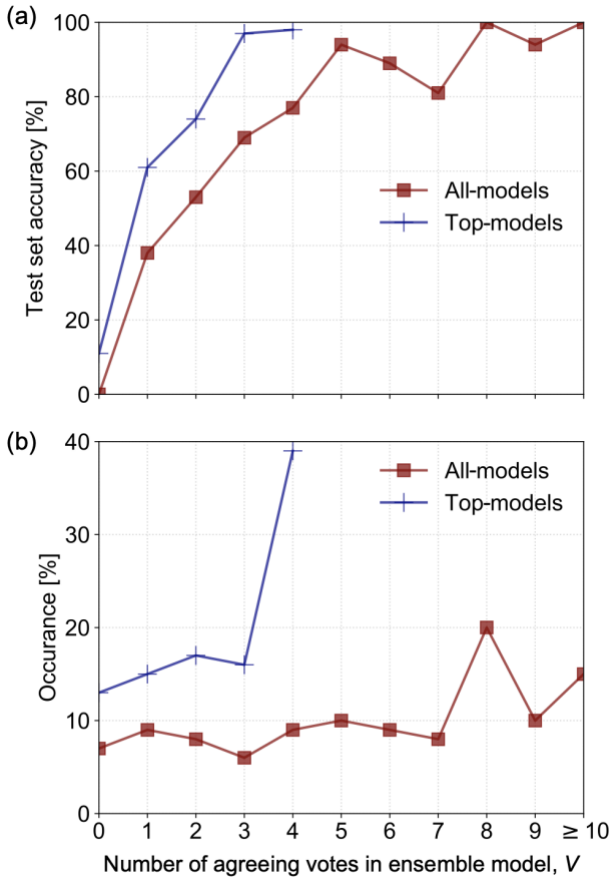


**Figure 10.** (a) The out-of-sample hand-drawn hydrocarbon recognition accuracy of the ensemble model when the top prediction has a given number of agreeing votes, *V*. (b) The percentage occurrence of a given number of agreeing votes for the top prediction. The accuracy is attributed to the confidence of the model when there are *V* votes for the top SMILES prediction.

## Conclusions

In this work, we demonstrate how deep learning can be used to develop an offline hand-drawn hydrocarbon structure recognition tool. We curated a large synthetic dataset and small hand-drawn dataset and explored how to best leverage the two to maximize the molecule recognition accuracy. The datasets were used to train an image-to-SMILES neural network to extract the molecule from a photographed hand-drawn hydrocarbon structure. By training with synthetic data only, we were able to achieve over 50% recognition accuracy on real-life hand-drawn hydrocarbons. We improved this accuracy by replacing 10% of the training set with augmented hand-drawn images and saw that applying fine-tuning resulted in a hand-drawn hydrocarbon recognition accuracy of nearly 70%. The trained data-driven models were combined with ensemble learning to achieve superior accuracy to the constituent models and gain

information on when the model would fail. The best ensemble model's top three predictions included the exactly correct molecule over 85% of the time, with over 50% of the predictions having over a confidence of nearly 100%. Extending the hydrocarbon recognition results presented in this paper to the recognition of all molecules offers an obvious extension, however, variation in hand-drawn font style and letter location provides a significant challenge.

Generative adversarial networks (GANs) have become popular in recent years for generating high quality synthetic data.[64-65] GANs were first introduced by Goodfellow et al. in 2014,[66] describing the idea of simultaneously training a *generative* model and a *discriminative* model. The generative model is trained to generate realistic training data, and the discriminative model is trained to distinguish between the generated data and the training data; by training the two models simultaneously, both are driven to improve at their respective roles. The result is a generated data distribution that matches the input data, and hence training data of the desired distribution can be produced for NN training. GANs can be thought of a highly sophisticated alternative to data augmentation, offering an exciting avenue to explore in future studies of hand-drawn molecule recognition.

The chemical structure recognition software developed in this work has many interesting use cases, such as connecting it to a user interface to be used as a phone or tablet application. A wide range of chemistry software could then be connected to the backend such as theoretical chemistry packages, lab notebooks and analytical tools. It would be particularly useful for software that currently requires knowledge of coding, command line scripting, and specialized input file format and so is inaccessible to large sections of the chemistry community. Since drawing a chemical structure by hand is a familiar task for all chemists, this app would lower the barrier of accessing such software. As a result, these currently unattainable tools could be readily incorporated into laboratories and classrooms to catalyse advances in chemical research and education.

**Supporting Information**

Details of image processing, neural network training, and example image predictions. Link to code to generate data and run training experiments.

**Acknowledgements**

This work was supported by the Office of Naval Research (N00014-18-1-2659).

## References


1. Rosenblatt, F., The perceptron: a probabilistic model for information storage and organization in the brain. *Psych. Rev.* **1958,** *65* (6), 386.
2. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y., *Deep learning*. MIT press Cambridge: 2016; Vol. 1.
3. Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C., Machine learning for molecular simulation. *Ann. Rev. Phys. Chem.* **2020,** *71*, 361-390.
4. Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A., Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **2012,** *108* (5), 058301.
5. Behler, J., Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.* **2016,** *145* (17), 170901.
6. Raucci, U.; Valentini, A.; Pieri, E.; Weir, H.; Seritan, S.; Martinez, T. J., Voice-controlled quantum chemistry. *Nature Comp. Sci.* **2021,** *1*, 42-45.
7. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P., Gradient-based learning applied to document recognition. *Proc. IEEE* **1998,** *86* (11), 2278-2324.
8. Bluche, T.; Louradour, J.; Messina, R. In *Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention*, 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), IEEE: 2017; pp 1050-1055.
9. Michael, J.; Labahn, R.; Grüning, T.; Zöllner, J. In *Evaluating sequence-to-sequence models for handwritten text recognition*, 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE: 2019; pp 1286-1293.
10. Graves, A.; Schmidhuber, J. In *Offline handwriting recognition with multidimensional recurrent neural networks*, Advances in neural information processing systems, 2009; pp 545-552.
11. Ingle, R. R.; Fujii, Y.; Deselaers, T.; Baccash, J.; Popat, A. C. In *A scalable handwritten text recognition system*, 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE: 2019; pp 17-24.
12. Plamondon, R.; Srihari, S. N., Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Patt. Anal. Mach. Learn.* **2000,** *22* (1), 63-84.
13. Rozas, R.; Fernandez, H., Automatic processing of graphics for image databases in science. *Journal of chemical information and computer sciences* **1990,** *30* (1), 7-12.
14. Contreras, M. L.; Allendes, C.; Alvarez, L. T.; Rozas, R., Computational perception and recognition of digitized molecular structures. *Journal of chemical information and computer sciences* **1990,** *30* (3), 302-307.
15. McDaniel, J. R.; Balmuth, J. R., Kekule: OCR-optical chemical (structure) recognition. *Journal of chemical information and computer sciences* **1992,** *32* (4), 373-378.
16. Casey, R.; Boyer, S.; Healey, P.; Miller, A.; Oudot, B.; Zilles, K. In *Optical recognition of chemical graphics*, Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93), IEEE: 1993; pp 627-631.
17. Ibison, P.; Jacquot, M.; Kam, F.; Neville, A.; Simpson, R. W.; Tonnelier, C.; Venczel, T.; Johnson, A. P., Chemical literature data extraction: the CLiDE Project. *Journal of Chemical Information and Computer Sciences* **1993,** *33* (3), 338-344.
18. Rajan, K.; Brinkhaus, H. O.; Zielesny, A.; Steinbeck, C., A review of optical chemical structure recognition tools. *J. Cheminf.* **2020,** *12* (1), 1-13.

19. Gkoutos, G. V.; Rzepa, H.; Clark, R. M.; Adjei, O.; Johal, H., Chemical machine vision: automated extraction of chemical metadata from raster images. *Journal of chemical information and computer sciences* **2003,** *43* (5), 1342-1355.
20. Rosania, G. R.; Crippen, G.; Woolf, P.; Shedden, K., A cheminformatic toolkit for mining biomedical knowledge. *Pharm. Res.* **2007,** *24* (10), 1791-1802.
21. Algorri, M.-E.; Zimmermann, M.; Friedrich, C. M.; Akle, S.; Hofmann-Apitius, M. In *Reconstruction of chemical molecules from images*, 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE: 2007; pp 4609-4612.
22. Valko, A. T.; Johnson, A. P., CLiDE Pro: the latest generation of CLiDE, a tool for optical chemical structure recognition. *J. Chem. Inf. Model.* **2009,** *49* (4), 780-787.
23. Filippov, I. V.; Nicklaus, M. C., Optical structure recognition software to recover chemical information: OSRA, an open source solution. ACS Publications: 2009.
24. Park, J.; Rosania, G. R.; Shedden, K. A.; Nguyen, M.; Lyu, N.; Saitou, K., Automated extraction of chemical structure information from digital raster images. *Chem. Cent. J.* **2009,** *3* (1), 4.
25. Park, J.; Saitou, K.; Rosania, G. In *Image-based automated chemical database annotation with ensemble of machine-vision classifiers*, 2010 IEEE International Conference on Automation Science and Engineering, IEEE: 2010; pp 168-173.
26. Sadawi, N. M.; Sexton, A. P.; Sorge, V. In *Chemical structure recognition: a rule-based approach*, Document Recognition and Retrieval XIX, International Society for Optics and Photonics: 2012; p 82970E.
27. Tharatipyakul, A.; Numnark, S.; Wichadakul, D.; Ingsriswang, S. In *ChemEx: information extraction system for chemical data curation*, BMC bioinformatics, Springer: 2012; p S9.
28. Frasconi, P.; Gabbrielli, F.; Lippi, M.; Marinai, S., Markov logic networks for optical chemical structure recognition. *J. Chem. Inf. Model.* **2014,** *54* (8), 2380-2390.
29. Beard, E. J.; Cole, J. M., ChemSchematicResolver: A Toolkit to Decode 2D Chemical Diagrams with Labels and R-Groups into Annotated Chemical Named Entities. *J. Chem. Inf. Model.* **2020,** *60* (4), 2059-2072.
30. Staker, J.; Marshall, K.; Abel, R.; McQuaw, C. M., Molecular Structure Extraction from Documents Using Deep Learning. *J. Chem. Inf. Model.* **2019,** *59* (3), 1017-1029.
31. Oldenhof, M.; Arany, A.; Moreau, Y.; Simm, J., ChemGrapher: Optical Graph Recognition of Chemical Compounds by Deep Learning. *arXiv preprint arXiv:2002.09914* **2020**.
32. Rajan, K.; Zielesny, A.; Steinbeck, C., DECIMER: towards deep learning for chemical image recognition. *J. Cheminf.* **2020,** *12* (1), 1-9.
33. Ouyang, T. Y.; Davis, R. In *Recognition of hand drawn chemical diagrams*, AAAI, 2007; pp 846-851.
34. Ramel, J.-Y.; Boissier, G.; Emptoz, H. In *Automatic reading of handwritten chemical formulas from a structural representation of the image*, Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318), IEEE: 1999; pp 83-86.
35. Sanchez-Lengeling, B.; Aspuru-Guzik, A., Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018,** *361* (6400), 360-365.
36. Elton, D. C.; Boukouvalas, Z.; Fuge, M. D.; Chung, P. W., Deep learning for molecular design—a review of the state of the art. *Mol. Sys. Design Eng.* **2019,** *4* (4), 828-849.

37. Weininger, D., SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **1988,** *28* (1), 31-36.
38. Kipf, T. N.; Welling, M., Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* **2016**.
39. Liu, Q.; Allamanis, M.; Brockschmidt, M.; Gaunt, A. In *Constrained graph variational autoencoders for molecule design*, Advances in neural information processing systems, 2018; pp 7795-7804.
40. Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A., SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry. *arXiv preprint arXiv:1905.13741* **2019**.
41. Jaeger, S.; Fulle, S.; Turk, S., Mol2vec: unsupervised machine learning approach with chemical intuition. *J. Chem. Inf. Model.* **2018,** *58* (1), 27-35.
42. Hirschberg, J.; Manning, C. D., Advances in natural language processing. *Science* **2015,** *349* (6245), 261-266.
43. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. In *Show and tell: A neural image caption generator*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015; pp 3156-3164.
44. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. In *Show, attend and tell: Neural image caption generation with visual attention*, International conference on machine learning, 2015; pp 2048-2057.
45. LeCun, Y.; Bengio, Y.; Hinton, G., Deep learning. *Nature* **2015,** *521* (7553), 436-444.
46. Krizhevsky, A.; Sutskever, I.; Hinton, G. E., Imagenet classification with deep convolutional neural networks. *Comm. ACM* **2017,** *60* (6), 84-90.
47. Hochreiter, S.; Schmidhuber, J., Long short-term memory. *Neur. Comp.* **1997,** *9* (8), 1735-1780.
48. Deng, Y.; Kanervisto, A.; Ling, J.; Rush, A. M. In *Image-to-markup generation with coarse-to-fine attention*, International Conference on Machine Learning, PMLR: 2017; pp 980-989.
49. Shi, B.; Bai, X.; Yao, C., An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Patt. Anal. Mach. Learn.* **2016,** *39* (11), 2298-2304.
50. Bahdanau, D.; Cho, K.; Bengio, Y., Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* **2014**.
51. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. In *Attention is all you need*, Advances in neural information processing systems, 2017; pp 5998-6008.
52. Genthial, G. Im2Latex. https://github.com/guillaumegenthial/im2latex.
53. Kingma, D. P.; Ba, J., Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
54. Abadi, M. i.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; Zheng, X. In *TensorFlow: A System for Large-Scale Machine Learning*, Savannah, GA, 2016/11//; {USENIX} Association: Savannah, GA, 2016; pp 265-283.
55. Fink, T.; Bruggesser, H.; Reymond, J. L., Virtual exploration of the small-molecule chemical universe below 160 daltons. *Ang. Chem. Int. Ed.* **2005,** *44* (10), 1504-1508.

56. Fink, T.; Reymond, J.-L., Virtual exploration of the chemical universe up to 11 atoms of C, N, O, F: assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery. *J. Chem. Inf. Model.* **2007,** *47* (2), 342-353.
57. Blum, L. C.; Reymond, J.-L., 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Amer. Chem. Soc.* **2009,** *131* (25), 8732-8733.
58. Bradski, G., The OpenCV Library. *Dr. Dobb's J. Soft. Tools* **2000**.
59. Wang, Q.; Gao, J.; Lin, W.; Yuan, Y. In *Learning from synthetic data for crowd counting in the wild*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; pp 8198-8207.
60. Kuznichov, D.; Zvirin, A.; Honen, Y.; Kimmel, R. In *Data augmentation for leaf segmentation and counting tasks in rosette plants*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019; pp 0-0.
61. Tajbakhsh, N.; Shin, J. Y.; Gurudu, S. R.; Hurst, R. T.; Kendall, C. B.; Gotway, M. B.; Liang, J., Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Trans. Med. Imag.* **2016,** *35* (5), 1299-1312.
62. Bishop, C. M., *Neural networks for pattern recognition*. Oxford university press: 1995.
63. Polikar, R., Ensemble learning. In *Ensemble machine learning*, Springer: 2012; pp 1-34.
64. Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; Ye, J., A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937* **2020**.
65. Goodfellow, I., NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* **2016**.
66. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. In *Generative adversarial nets*, Advances in neural information processing systems, 2014; pp 2672-2680.