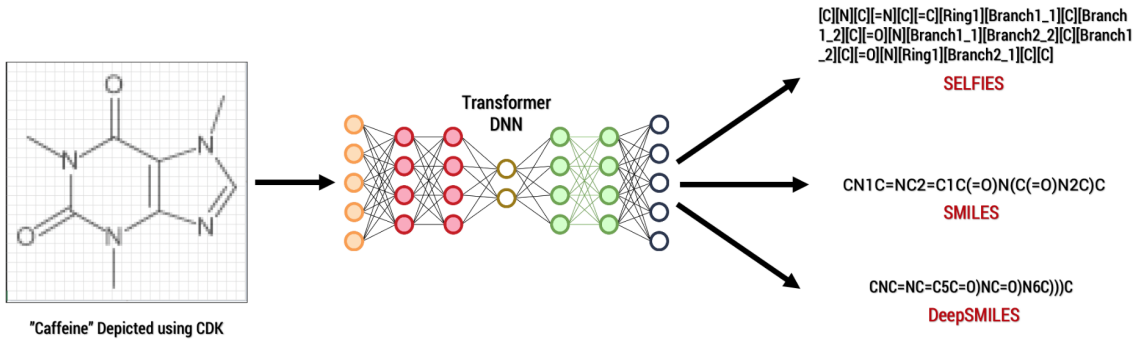
Performance of chemical structure string representations for chemical image recognition using transformers

Kohulan Rajan[1], Christoph Steinbeck[1] & Achim Zielesny[2*]

[1] Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany
[2] Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, D-45665 Recklinghausen, Germany
*Corresponding author email: achim.zielesny@w-hs.de

Abstract

The use of molecular string representations for deep learning in chemistry has been steadily increasing in recent years. The complexity of existing string representations, and the difficulty in creating meaningful tokens from them, lead to the development of new string representations for chemical structures. In this study, the translation of chemical structure depictions in the form of bitmap images to corresponding molecular string representations was examined. An analysis of the recently developed DeepSMILES and SELFIES representations in comparison with the most commonly used SMILES representation is presented where the ability to translate image features into string representations with transformer models was specifically tested. The SMILES representation exhibits the best overall performance whereas SELFIES guarantee valid chemical structures. DeepSMILES perform in between SMILES and SELFIES, InChIs are not appropriate for the learning task. All investigations were performed using publicly available datasets and the code used to train and evaluate the models has been made available to the public.



Keywords

Chemical data extraction, Deep learning, Neural networks, Optical chemical structure recognition, chemical string representations, DeepSMILES, SMILES, SELFIES

Introduction

Deep learning in chemistry is increasingly used to address problems in chemistry and cheminformatics [1]. One of these problems is Optical Chemical Structure Recognition (OCSR), which aims to decode a 2D bitmap image of a chemical structure into a computer-readable file or string representation. OCSR techniques are necessary, for example, to extract chemical structure information buried graphically in the chemical literature and patents [2] and store it in publicly available databases to enable their comprehensive retrieval with chemical structure, substructure or similarity searches. In a recent review paper, we surveyed the available OCSR tools, most of which rely on rule-based approaches [3–5] , and proposed deep learning solutions as a promising alternative [6].

OCSR approaches with deep learning utilize complex neural networks that require appropriate representations of chemical structures to encode and decode molecular information. Commonly, a 2D bitmap image of a chemical structure depiction is converted back into a textual representation - a character string - of that same structure. The human-readable SMILES [7] representation is one of the most widely used molecular string formats. But for deep learning purposes this line notation was shown to consist of a number of problems [8] which are primarily caused by the tokenization of its character string. As an example, structural branches are introduced with an opening bracket "(" and closed at a subsequent string position with a closing bracket ")". The same holds for ring openings and closures which are marked by a number where a ring opens or closes. However, once SMILES strings are partitioned into tokens based on characters, the precise placement of these markers at potentially distant positions within the text string causes problems for many deep neural networks. Due to these apparent inefficiencies new textual representations of chemical structures like DeepSMILES [8] and SELFIES [9] have recently been developed to overcome the sketeched problems. The DeepSMILES string representation aims at avoiding the problems due to branches in SMILES by using closing brackets only for branches where the number of brackets indicates the branch length. For ring closures a single symbol at the ring-closure location is used instead of two symbols at the ring-opening and ring-closing locations. In contrast to SMILES and DeepSMILES which have to be partitioned into single character tokens, the SELFIES representation defines separate enclosed tokens within square brackets "[...]" so that discrete meaningful tokens are provided by the representation itself (see Figure 1).

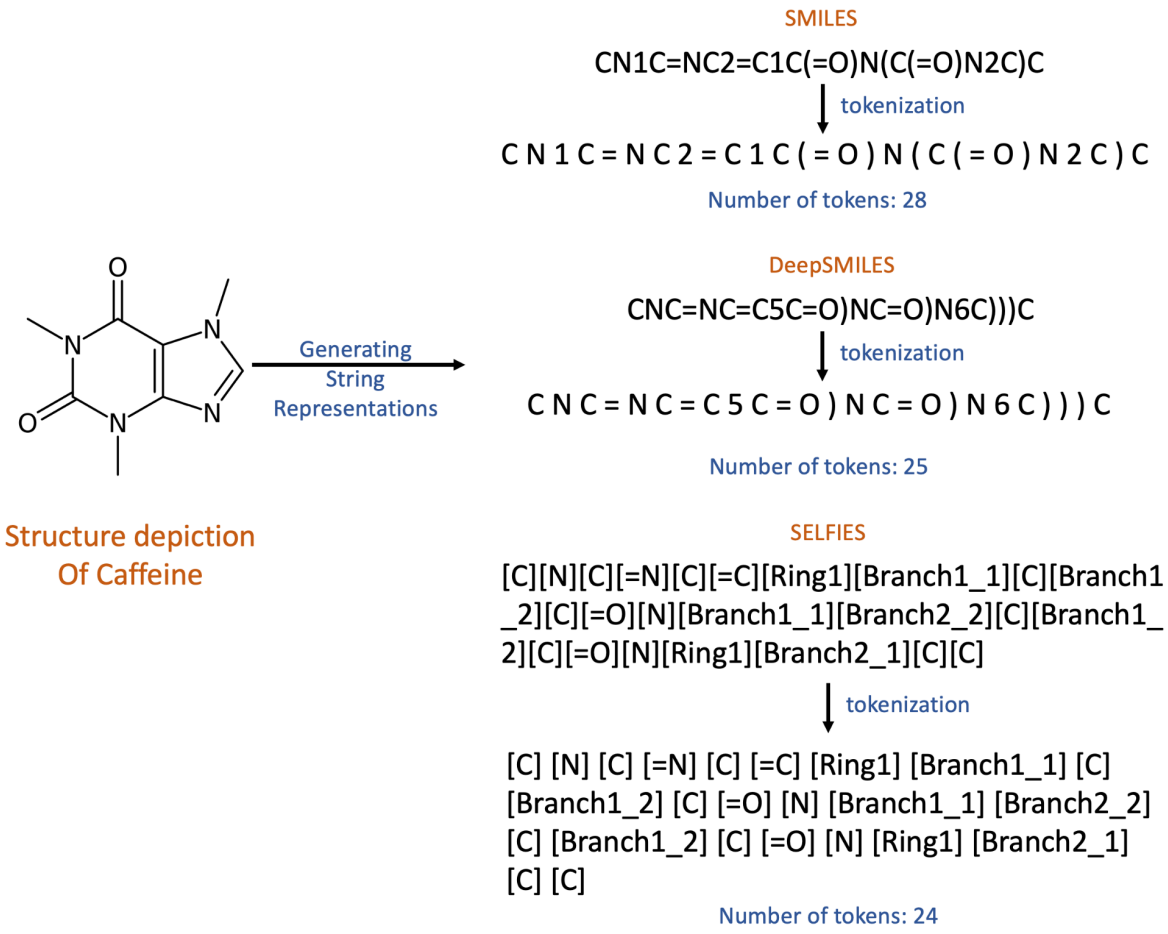


Figure 1: SMILES, DeepSMILES, and SELFIES are divided into tokens which are separated with spaces.

In a recent OCSR study [10], we encountered similar problems with SMILES representations which eventually led to a SELFIES based implementation. By using SELFIES as the output representation, a predicted SELFIES string always converts into a valid molecule due to the SELFIES decoding algorithm. In contrast, predicted SMILES may be invalid due to syntax errors such as mismatched binding symbols, branching, or ring closure. Other recent OCSR approaches [11–14] that used SMILES strings for output representation did not specifically address their inherent problems.

To further support OCSR development this work reports findings of a comparative case study for chemical image to chemical structure translation with SMILES, DeepSMILES and SELFIES. In addition, InChIs are included as an output which was proposed by a recent Kaggle competition [15].

Methods

Data

In this study, all data were taken from ChEMBL [16] and PubChem [17] databases. The data was originally downloaded in SDF format. Using the Chemistry Development Kit (CDK) [18] the chemical structures were converted into SMILES strings with and without stereochemistry information. After the SMILES conversion, the DECIMER filtering rules [10] were applied to obtain a balanced dataset. Then two datasets were created, one containing SMILES without stereochemistry and one with stereochemistry information.

The filtering rules for the datasets without stereochemistry included the following,

- have a molecular weight of fewer than 1500 Daltons,
- not possess counter ions,
- only contain the elements C, H, O, N, P, S, F, Cl, Br, I, Se and B,
- not contain isotopes of Hydrogens (D, T),
- have 3 - 40 bonds,
- only contain implicit hydrogens, except in functional groups,
- have less than 40 SMILES tokens,
- no stereochemistry was allowed.

After filtering, a total of 1,655,225 molecules were obtained from ChEMBL. Dataset partitioning into training and test datasets is a challenging task: With a simple random partitioning, the test dataset may not cover the relevant chemical space which could lead to biased results. To avoid this problem, the RDKit [19] MaxMin [20] algorithm was applied, so that equally diverse training and test subsets were created which cover a similar chemical space.

A set of 3 million molecules from PubChem was used to investigate whether the network performs better with more data. Here, the dataset was twice as large as the ChEMBL dataset. The PubChem dataset was filtered using the same rules as above, and the RDKit MaxMin algorithm was again applied to create the test set.

For the datasets with stereochemistry, a total of 1,653,833 molecules were obtained from ChEMBL and 3 million molecules from PubChem. Again, the RDKit MaxMin algorithm was used to select diverse training and test subsets. Table 1 provides an overview of the datasets.

The dataset with stereochemistry obtained from ChEMBL was a little smaller than the corresponding dataset without stereochemistry since stereochemistry adds new characters to SMILES, thereby lowering the number of available molecules due to the applied ruleset. With PubChem, however, the dataset size can be managed, since PubChem is much larger than ChEMBL.

Table 1: Overview of the datasets used in this study.

Database name	**ChEMBL**		**PubChem**	
Dataset name	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Dataset description	Without stereochemistry	With stereochemistry	Without stereochemistry	Without stereochemistry
Train dataset size	1536000	1536000	3072000	3072000
Test dataset size	119225	117833	250000	250000

To visualize the training and test dataset diversity, Morgan fingerprints [21] were generated using RDKit and a Principal Component Analysis (PCA) [22] was performed on the generated fingerprints, see Figure 1.

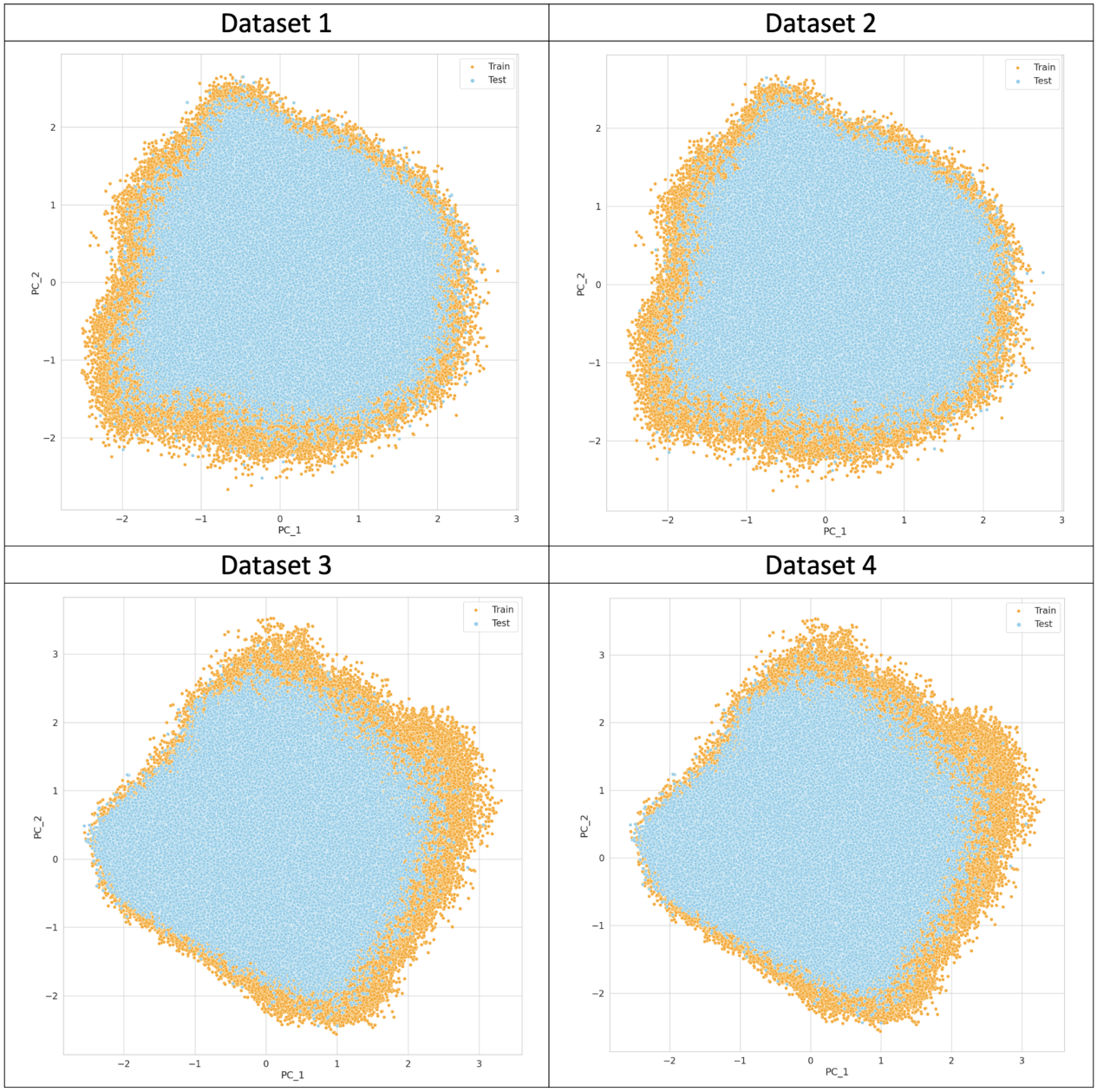


Figure 1: PCA plots visualising similar diversity of training and test datasets listed in Table 1.

Textual Data

The generated molecule sets were then converted into different textual representations of the chemical structures: SMILES, DeepSMILES, SELFIES and InChIs [23] and then split into tokens. For SELFIES this was a straightforward process since they already inherit a token-like word representation. Thus, SELFIES were split into tokens by using a space between the squared brackets “] [”.

For splitting SMILES, DeepSMILES and InChIs into tokens another set of rules had to be applied. They were split after,

- every heavy atom,
- every open bracket and close bracket “(”,”)”,
- every bond symbol “=”,”#”,
- every single-digit number and
- all the characters inside the squared brackets were retained as-is.

The "InChI=1S/" token was kept as one single token. As it is common in all InChIs, it was not used as a token during training but was later added to the predicted strings during post-processing to evaluate the results. The InChIs representation showed an inferior performance for the ChEMBL datasets compared to the other representations, so it was not included in the datasets generated from PubChem.
In addition to the token count, the maximum string length found in the datasets was calculated. This refers to the length of the longest string available in each dataset and also plays a role during training and testing. During training, the input vocabulary size which the network can handle was determined by comparing the number of tokens with the maximum length. In cases where the maximum length found in a dataset was smaller than the number of tokens available in the dataset, the input vocabulary size would be the number of tokens, otherwise, it would be the maximum length. During testing, the maximum length was used to determine when to stop predicting a structure if the end token is not met. Table 2 summarizes the number of tokens and the maximum string length found in each dataset. Datasets with stereochemistry information contain more tokens than datasets without. SELFIES representation led to more tokens than SMILES or DeepSMILES representation. InChIs had the lowest number of tokens but the largest maximum length of the longest string. With datasets 1 and 2, it became clear that InChIs perform significantly worse than the other string representations, so they were omitted in training and testing datasets 3 and 4.

Table 2: Overview of the token count and the maximum length.

Database name	ChEMBL				PubChem			
	Dataset 1 (Without stereochemistry)		Dataset 2 (With stereochemistry)		Dataset 3 (Without stereochemistry)		Dataset 4 (With stereochemistry)	
	Number of tokens	Maximum Length of String	Number of tokens	Maximum Length of String	Number of tokens	Maximum Length of String	Number of tokens	Maximum Length of String
SMILES	52	81	104	81	73	87	125	83
SELFIES	69	80	187	88	98	84	205	90
DeepSMILES	76	93	127	101	97	93	148	96
InChI	32	236	41	273	--	--	--	--

Image Data

A production-quality bitmap image of each molecule was generated with the CDK Structure Diagram Generator (SDG) at a resolution of 300x300 pixels. Each molecule was rotated by a random angle ranging from 0 to 360° and depicted. The generated images were saved in 8-bit PNG format. Each image contains a single structure only.

The features from these images were extracted as vectors by using the pre-trained weights of the 'noisy student' [24] trained EfficientNet-B3 [25] model. The extracted image features were then saved into NumPy arrays [26]. These topics were discussed in detail in our previous publication[27].

The extracted image features combined with the tokenized textual data were then converted into TFRecords [28]. TFRecords are binary records that can be used to train a model faster using Cloud Tensor Processing Units (TPUs) [29] on the Google Cloud Platform(GCP).

For training purposes, each TFRecord contains 128 data points consisting of 128 image feature vectors accompanied by 128 tokenized string representations. The TFRecords were generated on an in-house server and then moved into a Google Cloud Storage bucket.

Each dataset contains the same image data but different string representations.

Network, Training and Testing.

In this work, we use the same network as in [15], a transformer-based network model similar to the "Base model" as explained in Google's publication, *Attention Is All You Need* [30]. This network uses four encoder-decoder layers and eight attention heads. Attention has a dimension

size of 512 and feed-forward networks have a dimension size of 2048. The columns and rows here correspond to the image features we extracted as vectors, which are 10 × 10 × 1536. A dropout rate of 10% is used to prevent overfitting. According to the publication “Attention Is All You Need” the network is trained using the Adam optimizer with a custom learning rate scheduler. The loss is calculated by using sparse categorical cross-entropy between the real and predicted SELFIES. The network was coded with Python 3 using Tensorflow 2.3 [31] on the backend.

Throughout the training process, all models were trained on TPU v3-8 devices in the Google cloud. When comparing the training speed and network performance, a batch size of 1024 was found to be an adequate choice. The models were trained until the training loss had converged. In total, we trained eight models on datasets 1 and 2, and six models on datasets 3 and 4.

Once the models were fully converged, they were tested on an in-house server equipped with a GPU. To determine how many of the predictions were identical, the predictions were compared to the original strings. After the identical prediction calculations, all the predictions were converted to SMILES.

An analysis of the Tanimoto [32] similarity index was conducted between the original and predicted SMILES using PubChem fingerprints available in the CDK. The Tanimoto similarity indices help to understand how well the network was able to learn chemical string representations since sometimes the predictions were not identical but only similar to the original structures and even for isomorphic structures, there can be many different SMILES.

Results and Discussion

Transformer models can learn image and string representations more accurately and generalize well on unseen datasets. The purpose of this study is to examine different chemical string representations that are available for deep learning in chemistry and their performance on chemical image to string translation. Predictions were valid if the images get translated into structures correctly.

All the test results were assessed as following,

- Valid DeepSMILES/SELFIES/InChI: The predicted DeepSMILES, SELFIES and InChIs that could decode back into SMILES strings. The rest were deemed invalid.

- Valid SMILES: Predicted SMILES and decoded SMILES which could be parsed to calculate the Tanimoto similarity calculations. The rest were classified as invalid SMILES.

- Identical Predictions: This calculation identifies how many predictions match the original string representations. This was accomplished by using a one-to-one character string

match. If a single character was wrong in the predicted string, it was considered as a wrong prediction.

- Average Tanimoto: The Tanimoto similarity between the original and predicted SMILES was calculated from the valid SMILES and the average Tanimoto similarity index was calculated against the entire test dataset.

- Tanimoto 1.0 Percentage: The number of Tanimoto 1.0 counts on the calculated Tanimoto similarity indices of the valid SMILES and the percentage against the total test dataset.

Results for the ChEMBL dataset

From ChEMBL two datasets were obtained to train and test, one with stereochemistry (Dataset 1) and one without stereochemistry (Dataset 2). Table 3 summarizes the test results obtained with training on images from Dataset 1.

Table 3: Test results on dataset 1 (without stereochemistry)

	SMILES	**DeepSMILES**	**SELFIES**	**InChI**
Test dataset size	119225	119225	119225	119225
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.07%	0.00%	30.79%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.93%	100.00%	69.21%
Invalid SMILES	0.35%	0.10%	0.00%	0.00%
Valid SMILES	99.65%	99.83%	100.00%	69.21%
Identical Predictions (String match)	80.87%	78.67%	68.85%	64.28%
Tanimoto 1.0 Percentage (Not Identical)	86.30%	84.11%	73.88%	65.53%
Average Tanimoto	0.97	0.97	0.95	0.69

SMILES performed best in comparison to the other representations. Comparing the identical predictions and the Tanimoto 1.0 count, SMILES based models were more accurate. This was due to fewer tokens in the SMILES language space. Additionally, the maximum SMILES string was shorter than the rest. As a result, the model learns the representations better. Even though the InChIs have fewer tokens compared to the other representations, having a lesser number of tokens increases the maximum length of each string compared to the other representations,

which ultimately creates more errors for learning and predicting. In addition, valid InChI predictions were predominantly identical to the original string.
Even though SELFIES has the most valid structures, the overall predictivity of the SELFIES-based model was lower than that of SMILES and DeepSMILES. Overall, SMILES were more simple to learn - but for guaranteed valid structures, SELFIES were the best option.
To estimate the impact of stereochemistry, the same procedure was repeated with Dataset 2 where the models were trained from scratch. The results are summarized in Table 4.

Table 4: Test results on Dataset 2 (with stereochemistry)

	SMILES	**DeepSMILES**	**SELFIES**	**InChI**
Test dataset size	117833	117833	117833	117833
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.11%	0.00%	32.99%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.89%	100.00%	67.01%
Invalid SMILES	0.81%	0.64%	0.08%	0.00%
Valid SMILES	99.19%	99.25%	99.92%	67.01%
Identical Predictions (String match)	78.16%	77.07%	66.59%	59.10%
Tanimoto 1.0 Percentage (Not Identical)	85.02%	83.89%	72.07%	63.49%
Average Tanimoto	0.97	0.97	0.94	0.66

Based on the results given in Table 4, it was apparent that incorporating information on stereochemistry leads to a lowered accuracy. For DeepSMILES and InChIs, the number of invalid predictions increased. Additionally, the fraction of invalid SMILES increased for all representations except InChIs. After parsing all InChIs, there were only valid SMILES.
SMILES with stereochemistry reduced the overall predictability and accuracy due to the new artefacts added to the images. In addition, one should consider that the overall token count in these datasets increased due to stereochemistry so that additional tokens were introduced.
SMILES were overall best to get the most accurate predictions. Since InChIs showed a significantly inferior performance, it was decided to restrict further investigations to SMILES, DeepSMILES and SELFIES.

Results for the PubChem dataset

In order to determine how well the model can improve by increasing the number of data points, the size of the training and test data was doubled by utilizing data from PubChem. As pointed out above, InChIs were omitted in the subsequent testing.
The number of molecules available in PubChem is currently 110 million. For this work, 3 million molecules for training and 250,000 molecules for testing were obtained. In the obtained dataset,

the tokens were compared to those in the ChEMBL dataset to check we do have similar tokens present in the PubChem dataset. Using these datasets with and without stereochemistry (Datasets 3 and 4), the same training and testing procedure were repeated and the same evaluation procedure was used as before. For the dataset, without stereochemistry (Dataset 3) the results are summarized in Table 5.

Table 5: Test results on Dataset 3 (without stereochemistry)

	SMILES	DeepSMILES	SELFIES
Test dataset size	250000	250000	250000
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.08%	0.00%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.92%	100.00%
Invalid SMILES	0.22%	0.08%	0.00%
Valid SMILES	99.78%	99.84%	100.00%
Identical Predictions (String match)	88.62%	87.52%	82.96%
Tanimoto 1.0 Percentage (Not Identical)	92.19%	91.08%	86.42%
Average Tanimoto	0.98	0.98	0.97

By comparison of Table 5 with Table 3, it can be concluded that the data increase improved the model's performance in general. Again, SMILES show the best accuracy on test results and SELFIES still retain 100% valid structures.
DeepSMILES falls somewhere between these two. Although DeepSMILES has more valid structures than SMILES, when considering overall accuracy, the DeepSMILES format falls behind: comparing DeepSMILES to SELFIES, DeepSMILES has a better accuracy because of its SMILES like representation, but its overall number of valid structures lags behind SELFIES (see figure 2).

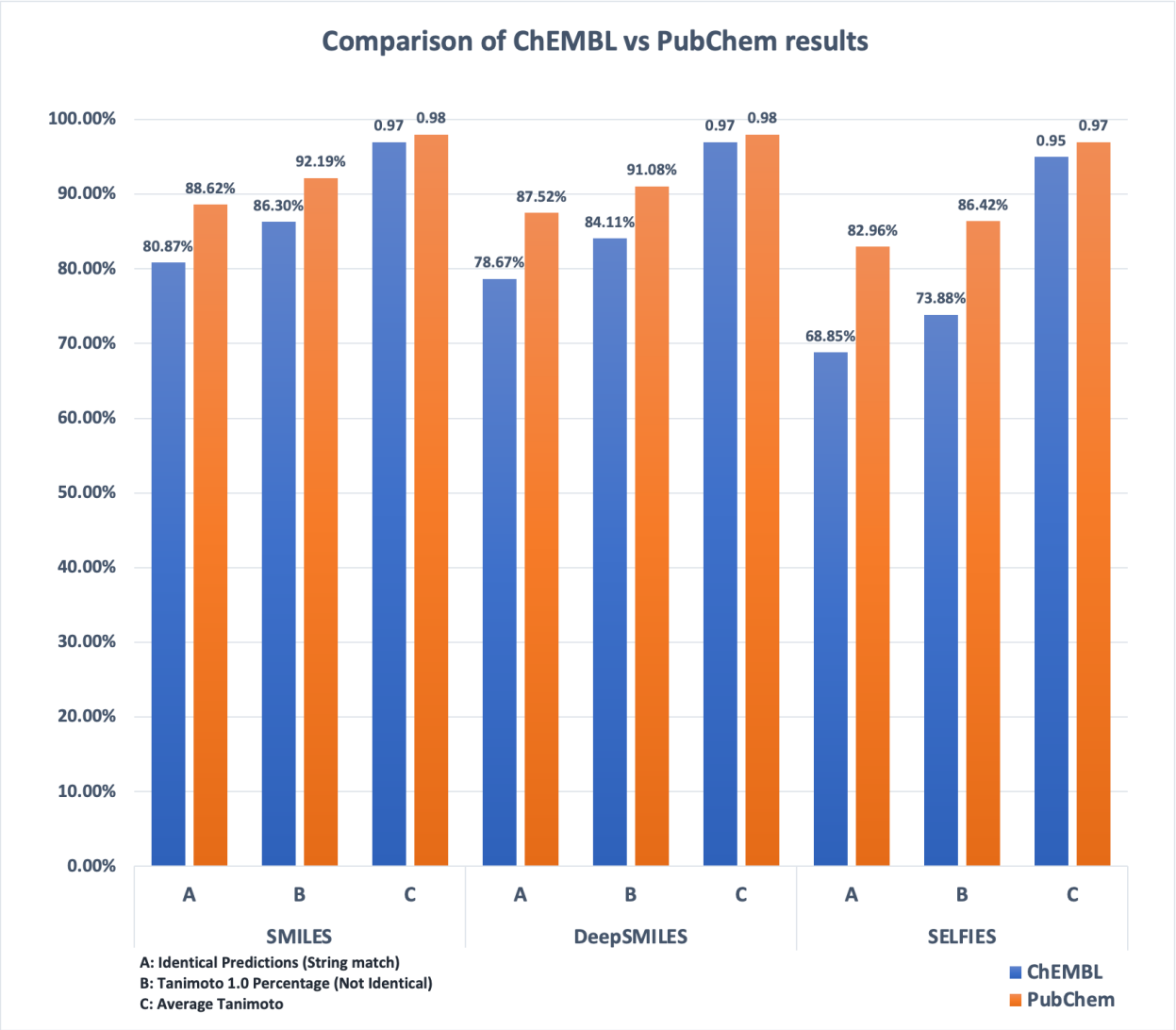


Figure 2: Comparison of identical predictions, Tanimoto 1.0 count and average Tanimoto of ChEMBL vs PubChem datasets (without stereochemistry).

A summary of the results for Dataset 4 with stereochemistry information can be found in table 6.

Table 6: Test results on Dataset 4 (with stereochemistry)

	SMILES	DeepSMILES	SELFIES
Test dataset size	250000	250000	250000
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.06%	0.00%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.94%	100.00%
Invalid SMILES	0.34%	0.05%	0.00%
Valid SMILES	99.66%	99.88%	100.00%
Identical Predictions (String match)	85.80%	83.80%	79.73%
Tanimoto 1.0 Percentage (Not Identical)	91.69%	90.60%	86.00%
Average Tanimoto	0.98	0.98	0.97

Compared to table 4, the results in table 6 show that increasing the dataset size does increase the overall accuracy where datasets with stereochemistry do not perform as well as datasets without stereochemistry. However, the overall accuracy does increase compared to the dataset from ChEMBL. In addition, all of the SELFIES predictions which were decoded back into SMILES are valid, providing 100% valid structures in comparison with table 4. SMILES perform best in terms of predictability and accuracy, see Figure 3.

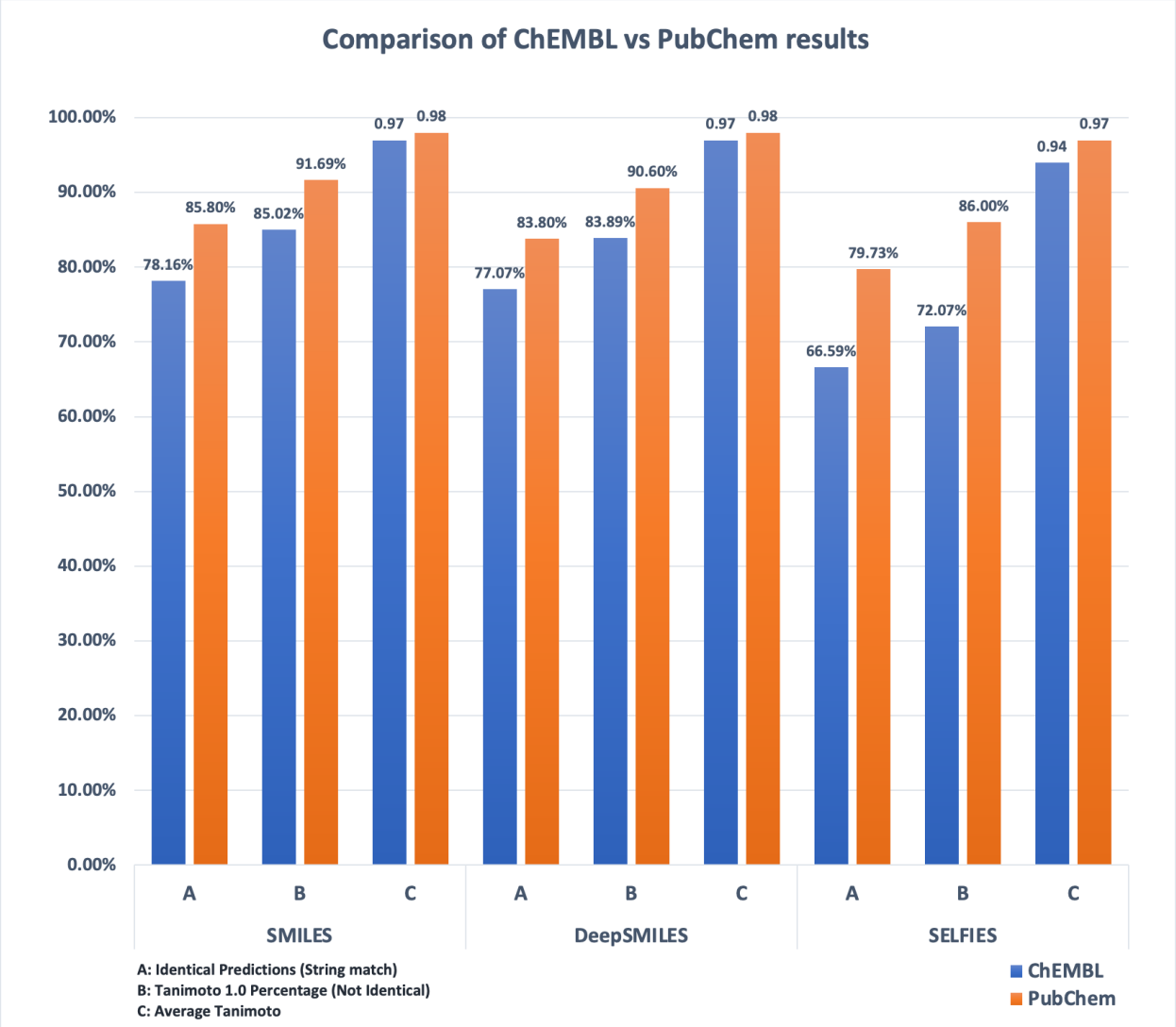


Figure 3: Comparison of identical predictions, Tanimoto 1.0 count and average Tanimoto of ChEMBL vs. PubChem datasets (with stereochemistry).

Conclusion

The performance of different textual chemical structure representations for the chemical image to structure translation using transformers was investigated. The most accurate models were obtained by using the SMILES representation. Using SELFIES, however, we were able to produce models that led to predictions with fewer invalid structures. DeepSMILES models always fell between SMILES and SELFIES. To ensure that the models perform similarly with more data, the datasets were scaled up. However, the results showed the same comparative performance. For most accurate predictions, models should be trained using SMILES, for maximizing valid structures SELFIES should be used.

The valid structures generated after decoding from SELFIES and DeepSMILES showed that the SELFIES decoding was superior to DeepSMILES decoding. SMILES and DeepSMILES should always be used with a set of rules on how to split them into meaningful tokens. SELFIES does not require this. There were fewer tokens in DeepSMILES than in SELFIES because the representation was similar to that in SMILES.

Since SELFIES encoding is a promising endeavour under active development, improved SELFIES variants could reach or even surpass the SMILES predictivity with the additional advantage of a 100% structural validity.

Availability of data and materials

The scripts are available at: https://github.com/Kohulan/DECIMER_Short_Communication
The data is available at: 10.5281/zenodo.5155037

Abbreviations

CDK - Chemistry Development Kit

DECIMER - Deep lEarning for Chemical ImagE Recognition

GCP - Google Cloud Platform

GPU - Graphical Processing Unit

InChI - International Chemical Identifier

PCA - Principal Component Analysis

SDF - Structure Data File

SDG - Structure Diagram Generator

SELFIES - Self-referencing embedded strings

SMILES - Simplified molecular-input line-entry system

TPU - Tensor Processing Units

Declarations

Competing interests

AZ is co-founder of GNWI - Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Funding

The authors acknowledge funding by the Carl-Zeiss-Foundation. Open Access funding enabled and organized by Projekt DEAL.

Authors' contributions

KR developed the software and performed the data analysis. CS and AZ conceived the project and supervised the work. All authors contributed to and approved the manuscript.

Acknowledgements

We are grateful for the company Google making free computing time on their TensorFlow Research Cloud infrastructure available to us.

References

1. Mater, A. C. & Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **59**, 2545–2559 (2019).
2. Tetko, I. V., Engkvist, O. & Chen, H. Does ‘Big Data’ exist in medicinal chemistry, and if so, how can it be harnessed? *Future Med. Chem.* **8**, 1801–1806 (2016).
3. Filippov, I. V. & Nicklaus, M. C. Optical structure recognition software to recover chemical information: OSRA, an open source solution. *J. Chem. Inf. Model.* **49**, 740–743 (2009).
4. Peryea, T., Katzel, D., Zhao, T., Southall, N. & Nguyen, D.-T. MOLVEC: Open source library for chemical structure recognition. in *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY* vol. 258 (AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA, 2019).
5. Smolov, V., Zentsev, F. & Rybalkin, M. Imago: Open-Source Toolkit for 2D Chemical Structure Image Recognition. in *TREC* (Citeseer, 2011).
6. Rajan, K., Brinkhaus, H. O., Zielesny, A. & Steinbeck, C. A review of optical chemical structure recognition tools. *J. Cheminform.* **12**, 60 (2020).
7. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to

methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).

8. O'Boyle, N. & Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. (2018) doi:10.26434/chemrxiv.7097960.v1.
9. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **1**, 045024 (2020).
10. Rajan, K., Zielesny, A. & Steinbeck, C. DECIMER: towards deep learning for chemical image recognition. *J. Cheminform.* **12**, 65 (2020).
11. Clevert, D.-A., Le, T., Winter, R. & Montanari, F. Img2Mol - Accurate SMILES Recognition from Molecular Graphical Depictions. *Chem. Sci.* (2021) doi:10.1039/D1SC01839F.
12. Khokhlov, I., Krasnov, L., Fedorov, M. & Sosnin, S. Image2SMILES: Transformer-based Molecular Optical Recognition Engine. *ChemRxiv* (2021) doi:10.26434/chemrxiv.14602716.v1.
13. Staker, J., Marshall, K., Abel, R. & McQuaw, C. M. Molecular Structure Extraction from Documents Using Deep Learning. *J. Chem. Inf. Model.* **59**, 1017–1029 (2019).
14. Weir, H. *et al.* ChemPix: automated recognition of hand-drawn hydrocarbon structures using deep learning. *Chem. Sci.* **12**, 10622–10633 (2021).
15. Bristol-Myers Squibb – molecular translation. https://www.kaggle.com/c/bms-molecular-translation.
16. Gaulton, A. *et al.* The ChEMBL database in 2017. *Nucleic Acids Res.* **45**, D945–D954 (2017).
17. Kim, S. *et al.* PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **47**, D1102–D1109 (2019).
18. Steinbeck, C. *et al.* The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **43**, 493–500 (2003).
19. Landrum, G. & Others. RDKit: Open-Source Cheminformatics Software.(2016). *URL*

http://www. rdkit. org/, https://github. com/rdkit/rdkit (2016).

20. Ashton, M. *et al.* Identification of diverse database subsets using property-based and fragment-based molecular descriptions. *Quant. struct.-act. relatsh.* **21**, 598–604 (2002).
21. Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **5**, 107–113 (1965).
22. Wold, S., Esbensen, K. & Geladi, P. Principal component analysis. *Chemometrics Intellig. Lab. Syst.* **2**, 37–52 (1987).
23. Heller, S. R., McNaught, A., Pletnev, I., Stein, S. & Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminform.* **7**, 23 (2015).
24. Xie, Q., Luong, M.-T., Hovy, E. & Le, Q. V. Self-training with noisy student improves imagenet classification. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 10687–10698 (2020).
25. Tan, M. & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. in *International Conference on Machine Learning* 6105–6114 (PMLR, 2019).
26. van der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering* **13**, 22–30 (2011).
27. Rajan, K., Zielesny, A. & Steinbeck, C. DECIMER 1.0: Deep Learning for Chemical Image Recognition using Transformers. (2021).
28. TFRecord and tf.train.Example. https://www.tensorflow.org/tutorials/load_data/tfrecord.
29. Norrie, T. *et al.* The Design Process for Google's Training Chips: TPUv2 and TPUv3. *IEEE Micro* **41**, 56–63 (2021).
30. Vaswani, A. *et al.* Attention Is All You Need. *arXiv [cs.CL]* (2017).
31. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv [cs.DC]* (2016).
32. Tanimoto, T. T. *An Elementary Mathematical Theory of Classification and Prediction*. (International Business Machines Corporation, 1958).