3D2SMILES: Translating Physical Molecular Models into Digital DeepSMILES Notations Using Deep Learning

Wenqi Marshall Guo*
Department of CMPS
University of British Columbia
Kelowna, BC
wg25r@student.ubc.ca

Yiyang Du
Department of CMPS
University of British Columbia
Kelowna, BC
duyiyang@student.ubc.ca

Mohamed Shehata
Department of CMPS
University of British Columbia
Kelowna, BC
mohamed.sami.shehata@ubc.ca

***Abstract*—Physical molecular models are widely used in educational settings for teaching organic and other branches of chemistry, offering an intuitive understanding of molecular structures. Conversely, while less intuitive, virtual models provide additional functionalities, such as retrieving molecular names and other properties. Currently, to the best of our knowledge, there is a gap between 3D molecular models and their digital counterparts. This paper introduces a computer vision model designed to bridge this gap by converting images of physical molecular models into their digital DeepSMILES representations. This conversion facilitates further information retrieval, enhancing educational utility. We developed synthetic and real datasets to train our model and evaluated its performance across various dataset combinations. Additionally, we attempted to improve the model's accuracy by multi-image input and beam search. We achieved 62.0% top-1 accuracy and 80.3% top-3 accuracy with beam search and multi-image input on our validation set. We also explored the model's characteristics, such as explainability by saliency maps, and examined its calibration. We also discussed the model's limitations and directions for future research.** [1]

I. Introduction

Molecular models are helpful in teaching chemistry because they make it easier for students to grasp the three-dimensional structure of molecules, unlike two-dimensional drawings. These models also help students visualize and think about molecules in three dimensions. A study [1] found that even though virtual and physical models are equally effective, students should use physical models early on. This is because people usually learn better with objects they can touch and see. Additionally, digital model tools have a learning curve, and it is hard for students to understand some space operations, such as a "chair flip" or bond rotation using digital models. However, digital model representation allows students to look up more information about the molecule, such as the IUPAC (International Union of Pure and Applied Chemistry) name, the boiling point, and the polarity. Students can also modify the molecule and observe how it affects its molecular properties. Therefore, it would be helpful if students could turn physical molecular models into digital ones.

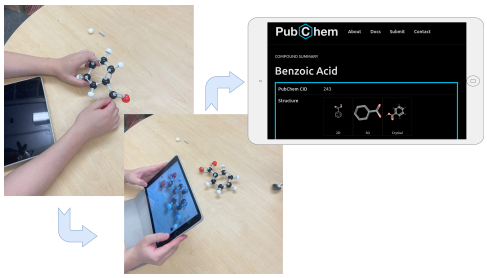

Fig. 1. **Designed Use Case:** This model is intended for integration with a mobile application interface; students could capture images of molecular structure representations, and the model would generate a corresponding SMILES string, which can then be used to redirect users to a detailed information page about the molecule.

Although there are models that can convert 2D molecular structure representations to corresponding digital representations [2][3][4], to the best of our knowledge, there is not a model for 3D molecular models. It will be beneficial to chemical education to close this gap between 3D molecular models and digital representations. Our contribution to this paper can be summarized as:

- We have constructed two datasets: one computer-rendered 3D molecular model and another consisting of real-world 3D molecular models.
- We trained a model on top of these two datasets capable of converting images of concrete molecular models into their respective SMILES (simplified molecular-input line-entry system) notations.
- We applied a method based on multi-image input and beam search, attempting to increase the accuracy of the output.

II. Related Work

To the best of our knowledge, no prior model can convert a 3D molecular set photo into its digital structure representation. However, there are some models for 2D structure formulas. The two most recent ones are SwinOCSR[2] and DECIMER.ai [3]. These two models are very similar in architecture, consisting of an image encoder and a transformer. The architectures

*The author is also affiliated with Weathon Software (https://weasoft.com).

[1]The code for this project and the two datasets are available on https://github.com/weathon/3d_mol_pub.

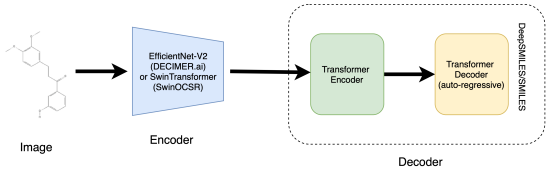

Fig. 2. **Model Architecture of DECIMER.ai and SwinOCSR:** The models receive a 2D molecular structure image as input. In the DECIMER.ai, an EfficientNet-V2 is used as the encoder, while SwinOCSR uses a Swin Transformer. Both models use a transformer encoder-decoder pair as the decoder for their models. The output from these models is either DeepSMILES or SMILES sequences. The output of this model is either DeepSMILES (for SwinOCSR) or SMILES sequences (for DECIMER.ai).

of DECIMER.ai and SwinOCSR are shown in Figure 2. ChemPix is another model that attempted to recognize hand-drawn hydrocarbon structures using CNN and Long Short-Term Memory (LSTM).

A. SwinOCSR

SwinOCSR [2] is a model that can translate printed molecular structures into DeepSMILES. It uses a Swin Transformer [5] as encoder and a transformer encoder-decoder pair as decoder [6]. SwinOCSR is trained with an image size of 224×224 with 4.5 million images, while both the validation and testing sets contain 250,000 images, each representing a total of 5 million molecules in their dataset. To address the imbalance of the tokens distribution (e.g. in the dataset, the token `C` (carbon) appears much more than = (double bond)), SwinOCSR uses focal loss [7] instead of normal cross-entropy loss. In their testing, the accuracy is 98.6% accuracy. However, on a real-world testing set, they only achieved 25.0% accuracy with a 0.5975 Tanimoto score.

B. DECIMER 1.0 And DECIMER.ai

Like SwinOCSR, DECIMER 1.0 [8] and DECIMER.ai [3] are also designed for reading molecule images and output as standard notations. DECIMER.ai is an improved version of the 1.0 version. Their main differences are encoder architecture, if the encoder is frozen, image size in the dataset, and different notation used. In the 1.0 version, the encoder is an EfficientNet-B3 [9] and is likely frozen during training. On the other hand, in DECIMER.ai, the encoder is an Efficient-V2-M [10]. Instead of the image size of 299×299 in the 1.0 version, DECIMER.ai uses 512×512 to provide sufficient information for bigger molecules. Based on their previous study, although DeepSMILES [11] has been proposed for machine learning models to generate more valid identifiers, SMILES lead to the most accurate identifiers, and SELFIES [12] lead to the most valid sequences, and DeepSMILES is between the two [13]. Thus, DECIMER.ai used SMILES instead of SELFIES like in DECIMER 1.0 [3].

In addition to printed structure formulas with regular shapes and formats, DECIMER.ai also attempted to recognize hand-drawn structures by finetuning the model with a dataset with augmented data that makes structures look hand-drawn. This increases sequence level accuracy from 27 to 60 and the Tanimoto similarity from 0.2 to 0.89.

C. ChemPix

ChemPix [4] is a model aimed at recognizing hand-drawn hydrocarbon (molecules with only carbon and hydrogen atoms, shown in structure formulas as mostly lines without letter notations) structures to their SMILES strings. They use a network architecture adapted from image captioning, with a CNN encoder and an LSTM with an attention mechanism and beam search as the decoder [4]. They limited their dataset to rings with sizes less than eight carbons, and molecules with multiple conjoined rings were not considered. They use both synthetic and authentic hand-drawn images to train the model. They also used a "voting committee" consisting of multiple models to vote for the final answer. In their testing, the model can reach 76% top-1 accuracy and 86% top-3 accuracy.

III. Method

We also used a two-stage training process in our project, similar to [3] and [4]. The model is first trained on a synthetic (3D rendered) dataset and then finetuned on a real-world collected dataset to reduce the amount of real-world data that needs to be collected as they are expensive.

A. Synthetic Dataset

We downloaded a set of compounds from PubChem [14], and then these compounds were filtered based on heavy atoms and molecular weights and only contained elements of C, O, S, H, N, Cl, Br, F, and P. Because of the complex 3D nature of molecules, there could be occlusions between atoms and bonds, where vital information is hidden in the image. Although there are methods to avoid occlusions by some spatial algorithm or human selection, we removed 3D spatial information in the pre-training stage. Thus, we downloaded the 2D coordinates from PubChem instead of 3D. This also makes the model focus on identifying atoms and their connections instead of the spatial information. After filtering, there are 79,369 compounds in the dataset. We created a Blender Python Script to render these compounds into 3D molecular images. OpenAI ChatGPT was used to write the building blocks (API Wrapper) and some script logic. The colour of each atom followed the CPK colouring convention and was according to a couple of online sources and 3D molecular models on the market. The atoms are rendered using spheres, single bonds are rendered using cylinders, and double bonds are rendered using tori. For each molecule, we rendered four different images at different angles. We use the image size of 224×224 given that our molecules are not too complex, and by using 224×224, we can better utilize the pre-training weights from the SwinOCSR encoder.

B. Real-World Dataset

We developed a mobile app using React.js for data collection, which displays a 3D interactive model preview [15] and allows users to capture images of their constructed models.

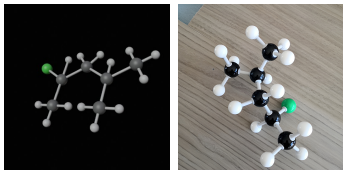

Fig. 3. **A rendered image and a real captured image of 2-Chloro-4-methyl pentane:** The left one is the synthetic image, and the right one is the real-world captured image

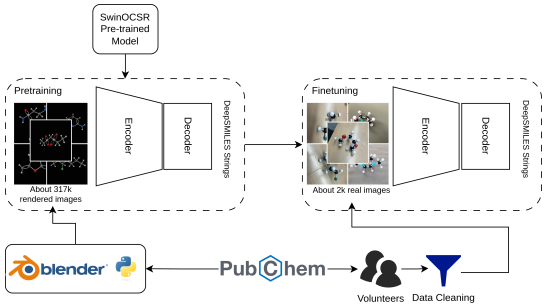

Fig. 4. **Training Method Overview:** The model is initialized using weights from SwinOCSR. It is first pre-trained on a synthetic dataset rendered by Blender using molecules from PubChem, then finetuning on a real dataset collected by volunteers.

We used GPT to group similar models based on a list of molecule names provided via the OpenAI API to enhance data collection efficiency. Users can build either predefined or chosen molecules. Due to the absence of sulphur in some model kits, sulphur-containing molecules were excluded. Volunteers were recruited to photograph these molecular models. Post-collection, issues like obstructions and motion blur in images prompted the creation of a secondary app for volunteers to review and filter images. Images were captured against clear backgrounds from various angles. We amassed 1934 images of 203 molecules, sorting them into training and validation sets based on SMILES strings to ensure no overlap. The validation set, chosen randomly, consists of 29 molecules and 258 images.

$$P(y_{t,j}|y_0 \dots y_{t-1}, I) = \frac{1}{|I|} \sum_{i \in I} P(y_{t,j}|y_0 \dots y_{t-1}, i) \quad (1)$$

IV. Experiments and Results

A. Model Architecture

Our model architecture is adapted from SwinOCSR [2] with minimal changes. We also used Focal loss [7] for our loss function like in [2]. During training, we use the "teacher forcing" method. Unlike in other natural language models, we did not use scheduled sampling [16] (designed to let the model learn to recover from mistakes). Because unlike natural language, which has some variation and fault tolerance, the DeepSMILES we generated has only one correct answer. Thus, it does not matter if the error is cumulative, as the DeepSMILES would be wrong either way.

B. Beam search and Multi-image input

Beam search is a way to increase the accuracy of autoregressive models [17] [18]. Beam search can also provide users with multiple candidate answers to choose from, similar to the top-n in image classification. However, one issue with beam search is that it will favour shorter sequences as a sub-sequence will always have a higher probability than its parent sequence [17] [18]. There are multiple methods to solve this issue. Some are mentioned in [17] and [18]. In this paper, we use a simple method mentioned in [18] where we multiply $\frac{1}{L^\alpha}$ to the log probability where L is the length of the sequence. This will make the log probability of a long sequence higher (since the log probability is negative). It can also be seen as taking the mean of the $logprob$ of each token when $\alpha = 1$. The re-scored $logprob$ is shown in Equation 2 [18] where $P(y_t|y_1 \dots y_{t-1}, \mathbf{e})$ is the probability of output y_t at t-th token given previous tokens $y_1 \dots y_{t-1}$ and image embedding $\mathbf{e}$.

$$LP = \frac{1}{L^\alpha} \sum_{t=1}^{L} logP(y_t|y_1 \dots y_{t-1}, \mathbf{e}) \quad (2)$$

We also hypothesize that multiple image inputs will increase the model's accuracy. This has been shown in previous work [19], in which the authors use the "early fusion" method where feature maps from 3 different images are concatenated on the channel axis in the early stage of the network. However, this method requires modification of the backbone network, and the number of images used in this method is fixed. Instead, our method passes individual images through the entire network and averages the softmax distribution of tokens, as shown in Equation 1 where I is the input image set, M is the model, $P(y_{t,j})$ is the probability of token j at time t, and σ is the softmax function. This is similar to the multi-crop technique used since the early days of applying deep networks in computer vision, like in AlexNet [20] and ResNet [21].

We train our model on the synthetic dataset with AdamW [22] and an exponential learning rate scheduler. With the fixed validation set, we did a hyperparameter search on the finetuning dataset using Weights and Biases [23] to select models with the highest validation token-level accuracy. All accuracy reported is the last validation accuracy regardless of oscillating.

C. Training Stages

Our model has been trained on three datasets: 1) printed chemical structure diagrams to SMILES strings conversion with 5 million molecules from PubChem by SwinOCSR [2], 2) our synthetic dataset containing about 320K images of 80K molecules (Synt-80K), 3) and our real dataset with about 2K images for 200 molecules. We hypothesize that each stage benefits the final model in terms of performance. We perform an ablation study on these stages. For training stages involving synthetic-80k pre-training, we do not do much hyperparameter search as long as the model converges at a reasonable rate and reaches 99.7% validation token level accuracy. We used hyperparameter optimizers for training involving the real-200

dataset for learning rate, epoch, and learning rate decay. The hyperparameter is searched for each setting independently. With teacher forcing, we used a fixed validation for around 30 trials on token-level validation accuracy.

The results of this experiment are shown in Table I. We observe that when the Synt-80k dataset is used, the sequence level accuracy (SLA) is around 60%. However, when it is not used, the SLA drops dramatically to between 0% and 5%, which will be unusable in any real-world setting. Interestingly, the choice of pre-training for the encoder (either ImageNet or SwinOCSR) does not seem to affect the results significantly.

Case	Encoder Pre-training	SwinOCSR Decoder Wt.	Synt-80K	TLA	SLA
Ours	SwinOCSR	✓	✓	95.5%	64.9%
(a)	ImageNet	✓	✓	94.9%	62.5%
(b)	SwinOCSR	✓	×	71.7%	5.43%
(c)	None	×	×	60.4%	0.38%

TABLE I

TOKEN LEVEL ACCURACY (TLA) AND SEQUENCE LEVEL ACCURACY (SLA) OF THE SWIN-L MODEL: THE MODEL WAS TRAINED USING VARIOUS COMBINATIONS OF DATASETS. THE ENCODER COULD BE TRAINED ON THE SWINOCSR DATASET, IMAGENET, OR WITHOUT ANY PRE-TRAINING. THE DECODER COULD BE TRAINED ON SWINOCSR OR WITHOUT ANY PRE-TRAINING. THE ENTIRE MODEL COULD THEN BE TRAINED WITH OR WITHOUT THE SYNTHETIC-80K DATASET, AND ALL CONFIGURATIONS WERE FINALLY FINETUNED TO THE REAL-2K DATASET.

D. Beam Size and Image Counts

We tested the performance of our model with varying beam sizes and image counts. The experiment is performed on Swin-L with an image size of 224^2 model using *ours* model from the last section. A grid search was applied to beam size, number of image inputs, alpha (the penalty for beam search mentioned in Eq. 2), and temperature for Softmax. The results are depicted in Figure 5. For each hyperparameter set, we iterate through all the molecule IDs and for each ID, we select 200 combinations, with each randomly selecting a designated number of images without replacement and get the average accuracy[2].

The figure shows that when the beam size is set to 1, the top-1 accuracy slightly increases as the number of image inputs rises, likely because multiple images provide more information. However, the beam size did not consistently enhance the top-1 accuracy. We suspect this is due to the low calibration of our model, which will be further discussed in the discussion section.

For the top-n accuracy, where n equals the number of beams, we observe that increasing either the number of images or the number of beams consistently increases the accuracy. Notably, with a beam size of 3 and 3 image inputs, the model achieves an impressive 80.3% accuracy, a significant improvement compared to 62.0%.

The increases of top-n accuracy with beam size are not observed with the top-1 accuracy. We also suspect this is due to the model's low calibration.

[2]Some runs crashed during running, which results in less than 200 runs for them

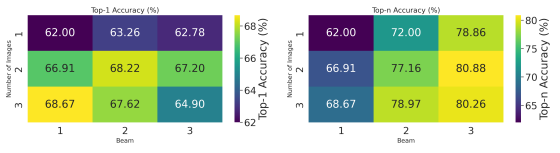

Fig. 5. **The model's accuracy with beam numbers and number of images input:** This figure illustrates the relationship between beam size and the number of input images with the model's performance. The top-1 accuracy represents the accuracy of the model's output with the highest log probability. In contrast, the top-n accuracy ($n = beam$) indicates the model's accuracy within the given beam size.

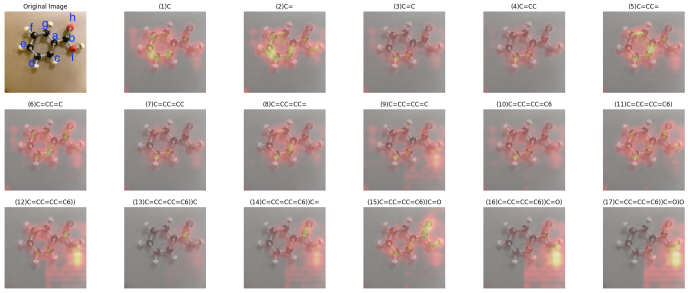

Fig. 6. **Saliency Map for an Input Image of Benzoic Acid:** The map illustrates the model's focus when it generates the last token of the sequence in the subtitle. Brighter regions indicate higher absolute values of the partial derivatives, highlighting which part the model determines as important for the decision of the token.

V. DISCUSSIONS

A. Explainability

Saliency map [24] is one common way to explain the behaviours of image networks. It takes the absolute value of the partial derivative of the input image with respect to the target output element. (Here it is the argmax element in the last token of the generated sequence) To show the saliency for better visualization, we take the square root of the partial derivative with blurring effects. Eq. 3 shows the saliency map S as a function of input image x, and L is the current sequence length.

$$S(x) = \sqrt{|\frac{\partial x}{\partial M(x)_{L,j}}|}, \textbf{where} j = \arg\max M(x)_{L,:} \quad (3)$$

Figure 6 shows the saliency map of the model with an input of benzoic acid. Overall, these maps make sense for the output. For instance, in (13), the saliency map has a very bright spot on the hydroxyl carbon (Carbon b), which matches the carbon it identifies where it is connected to the double bond with the Oxygen (h) and the ring. Also, in (14), the model successfully highlights the double bond between Carbon (b) and Oxygen (h). In (17), it accurately highlights Oxygen (i). However, there are also some less optimal cases. In (16), (12), and (17), the saliency map shows a very bright region in the bottom right corner where there is no atom, and in (14), (12), and (9), there is a large block region in the bottom right corner.

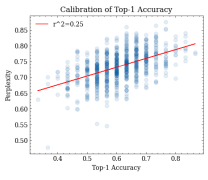

Fig. 7. **Calibration:** The calibration figure of the model's top-1 sequence level accuracy and corresponding perplexity. This shows how well the model can predict if its answer is correct.

B. Error Analysis

We collected all mistakes the model made, and we categorized the errors into one of these categories: wrong atom (wrong atom type or wrong number of atoms), branch/ring mistake (branch or ring closure at the wrong location), end of sequence (end the sequence too early or too late) error, bond order error (e.g. mistaken triple bond with single bond), and others. Table II shows examples of each error, with the wrong atom being the most common error.

C. Calibration

The model's calibration is the metric determining if the model's output accuracy is aligned with its uncertify. The calibration metric of a model is important when it needs to make important decisions such as resume screaming and malware detection [25]. The calibration is important for our sequence-generating task for another reason: we will select the output with the lowest uncertainty (highest *logprob*) as the final output. We evaluated our model's calibration using a method similar to [25], we randomly sample in each sample, we randomly select 30 molecules with replacement and then randomly choose images for each molecule. We calculate the mean accuracy for each sample and the mean perplexity ($logprob$) for each sample and plot them in Figure 7. Because the sequence probability (perplexity) is the product of the probabilities of each token, the value is usually not normalized in [0,1] range. Thus, it did not use the commonly used metric expected calibration error (ECE) like in [25]. Even in [25], they did not evaluate the calibration for generative tasks. Instead, we used the r^2 value to evaluate the calibration of the model. This is partially inspired by the r^2 values used in evaluating the quality of the calibration curve in analytical chemistry.

In the figure, we can observe that the r^2 value is relatively small (0.25). This low correlation could explain why the model's top-1 accuracy did not significantly improve as the beam size increased, despite the increase in top-n accuracy: even with more options available to choose from, the model lacks a reliable method to determine which answer is correct and rank it at the top.

VI. Conclusion

This paper presents a model designed to convert physical molecular models into their digital DeepSMILES notation, aiming to close the gap between physical models and virtual representations. We conducted experiments on various dataset combinations and the effects of beam size, number of images input, and dataset size. We also investigated some interesting properties of our model, such as the model's explainability, through saliency maps and performed error analysis. We also proposed our hypothesis about why beam size did not improve top-1 accuracy significantly, as expected, by examining the model's calibration.

VII. Limitation and Future Work

In this project, we did not attempt to recognize stereochemical information such as chirality and E/Z structure. However, these are important concepts in organic chemistry for students to understand and being able to show these is one of the key characteristics and benefits of 3D models. However, this is challenging because it requires the computer vision model not only to understand the connectivity of the atoms but also their relative positions. There are minimal differences between the two stereoisomers. Additionally, the model needs to learn how to output SMILES with stereochemical information (such as iso-SMILES) rather than regular SMILES.

To encourage the model to differentiate between molecules with different stereochemical configurations, we can use contrastive learning. Given images M of a molecule, we have n images representing one stereochemical configuration ($M_a \subset M, |M_a| = n$) and m images representing another stereochemical configuration ($M_b \subset M, |M_b| = m$). The model is trained to cluster images within the same set (M_a or M_b) closely while pushing apart the clusters corresponding to the two different sets. Because the iso-SMILES between two stereochemical configurations could be too similar to motivate the model to distinguish them, adding this contrastive loss could encourage the model to learn the differences.

To improve the calibration of the model, we can use other types of confidence scores rather than the $logprob$, such as self-evaluation [26]. That is, at the end when the model outputs the `<end>` token, itself or another side model outputs a confidence score directly. This could be a better metric than $logprob$ for final selection ($logprob$ is still used in beam search).

The size of our Real-200 dataset is constrained by time and budget. Our preliminary, non-rigorous experiment suggests that increasing the number of molecules in the dataset and the number of images per molecule could potentially enhance performance. It would be worthwhile to investigate the model's performance with a larger dataset in future studies.

Our model can already run in a few seconds on modern hardware. Distillation or quantization could be used to further optimize the speed.

Additionally, we only proposed a model that could potentially used in chemistry classrooms; however, we did not test its educational value. Future chemistry education research is needed to understand its effectiveness and the improvement required.

Acknowledgement

The creation of this project was aided by OpenAI ChatGPT and Google Gemini, which contributed to tasks including but not limited to brainstorming, troubleshooting, writing feedback, and coding support.

The computation of this work is partially performed on UBC Advanced Research Computing, DOI: 10.154288/SOCKEYE. This research was partly enabled by support provided by BC DRI Group and the Digital Research Alliance of Canada (alliancecan.ca).

Author Contribution

The first author proposed the idea, finished the first version of the paper, drafted the paper, and conducted the experiments. The second author drafted the literature review, helped edit the paper,

Error Type		Wrong Atom(s)	Branch/Ring	End of Sequence	Bond Order	Others
Example	Correct	CCCC4)Cl	CCC)C)CC)C)C	CNCCCCCCNC	CCNCC#N	CCCC4)Cl
	Pred	CCCCl	CCC)CC)C	CNCCCCCCNC)C	CCNCCN	C=CCl
Counts		40	26	10	8	12

TABLE II

SUMMARY OF ERROR TYPES WITH EXAMPLES: WE CATEGORIZE THE MISTAKES MADE BY THE MODEL DURING INFERENCE INTO THESE FIVE CLASSES.

and conducted a few experiments with debugging. The last author supervised the project and provided funding and resources.

REFERENCES

[1] V. F. Savec, M. Vrtacnik, and J. K. Gilbert, "Evaluating the Educational Value of Molecular Structure Representations," en, in *Visualization in Science Education*, ser. Models and Modeling in Science Education, J. K. Gilbert, Ed., Dordrecht: Springer Netherlands, 2005, pp. 294–295, ISBN: 9781402036132. DOI: 10.1007/1-4020-3613-2_14. [Online]. Available: https://doi.org/10.1007/1-4020-3613-2_14 (visited on 01/24/2024).

[2] Z. Xu, J. Li, Z. Yang, S. Li, and H. Li, "SwinOCSR: End-to-end optical chemical structure recognition using a Swin Transformer," *Journal of Cheminformatics*, vol. 14, no. 1, p. 41, Jul. 2022, ISSN: 1758-2946. DOI: 10.1186/s13321-022-00624-5. [Online]. Available: https://doi.org/10.1186/s13321-022-00624-5 (visited on 08/06/2023).

[3] K. Rajan, H. O. Brinkhaus, M. I. Agea, A. Zielesny, and C. Steinbeck, *DECIMER.ai - An open platform for automated optical chemical structure identification, segmentation and recognition in scientific publications*, en, Feb. 2023. DOI: 10.26434/chemrxiv-2023-xhcx9. [Online]. Available: https://chemrxiv.org/engage/chemrxiv/article-details/63efb90dfcfb27a31ff39b08 (visited on 08/09/2023).

[4] H. Weir, K. Thompson, A. Woodward, B. Choi, A. Braun, and T. J. Martínez, "ChemPix: Automated recognition of hand-drawn hydrocarbon structures using deep learning," English, *Chemical Science*, vol. 12, no. 31, Jul. 2021, ISSN: 2041-6520. DOI: 10.1039/d1sc02957f. [Online]. Available: https://www.osti.gov/pages/biblio/1817872 (visited on 08/06/2023).

[5] Z. Liu, Y. Lin, Y. Cao, *et al.*, *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*, en, Mar. 2021. [Online]. Available: https://arxiv.org/abs/2103.14030v2 (visited on 08/09/2023).

[6] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention Is All You Need*, arXiv:1706.03762 [cs], Aug. 2023. [Online]. Available: http://arxiv.org/abs/1706.03762 (visited on 08/06/2023).

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection*, Feb. 7, 2018. arXiv: 1708.02002[cs]. [Online]. Available: http://arxiv.org/abs/1708.02002 (visited on 04/23/2024).

[8] K. Rajan, A. Zielesny, and C. Steinbeck, "DECIMER 1.0: Deep learning for chemical image recognition using transformers," *Journal of Cheminformatics*, vol. 13, no. 1, p. 61, Aug. 2021, ISSN: 1758-2946. DOI: 10.1186/s13321-021-00538-8. [Online]. Available: https://doi.org/10.1186/s13321-021-00538-8 (visited on 07/08/2024).

[9] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," Sep. 2020, arXiv:1905.11946 [cs, stat]. DOI: 10.48550/arXiv.1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946 (visited on 03/02/2024).

[10] M. Tan and Q. V. Le, *EfficientNetV2: Smaller Models and Faster Training*, en, Apr. 2021. [Online]. Available: https://arxiv.org/abs/2104.00298v3 (visited on 08/09/2023).

[11] N. O'Boyle and A. Dalke, *DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures*, en, Sep. 2018. DOI: 10.26434/chemrxiv.7097960.v1. [Online]. Available: https://chemrxiv.org/engage/chemrxiv/article-details/60c73ed6567dfe7e5fec388d (visited on 07/09/2024).

[12] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation," *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045 024, Dec. 2020, arXiv:1905.13741 [physics, physics:quant-ph, stat], ISSN: 2632-2153. DOI: 10.1088/2632-2153/aba947. [Online]. Available: http://arxiv.org/abs/1905.13741 (visited on 07/09/2024).

[13] K. Rajan, C. Steinbeck, and A. Zielesny, "Performance of chemical structure string representations for chemical image recognition using transformers," en, *Digital Discovery*, vol. 1, no. 2, pp. 84–90, Apr. 2022, ISSN: 2635-098X. DOI: 10.1039/D1DD00013F. [Online]. Available: https://pubs.rsc.org/en/content/articlelanding/2022/dd/d1dd00013f (visited on 07/09/2024).

[14] S. Kim, J. Chen, T. Cheng, *et al.*, "PubChem 2023 update," en, *Nucleic Acids Research*, vol. 51, no. D1, pp. D1373–D1380, Jan. 2023, ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkac956. [Online]. Available: https://academic.oup.com/nar/article/51/D1/D1373/6777787 (visited on 08/04/2023).

[15] N. Rego and D. Koes, "3Dmol.js: Molecular visualization with WebGL," en, *Bioinformatics*, vol. 31, no. 8, pp. 1322–1324, Apr. 2015, ISSN: 1367-4811, 1367-4803. DOI: 10.1093/bioinformatics/btu829. [Online]. Available: https://academic.oup.com/bioinformatics/article/31/8/1322/213186 (visited on 04/30/2024).

[16] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, *Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks*, arXiv:1506.03099 [cs], Sep. 2015. [Online]. Available: http://arxiv.org/abs/1506.03099 (visited on 04/30/2024).

[17] Y. Yang, L. Huang, and M. Ma, *Breaking the beam search curse: A study of (Re-)scoring methods and stopping criteria for neural machine translation*, arXiv:1808.09582 [cs], Oct. 2018. [Online]. Available: http://arxiv.org/abs/1808.09582 (visited on 02/08/2024).

[18] A. Zhang, Z. Lipton, M. Li, and A. J. Smola, "Beam Search," eng, in *Dive into deep learning*, Cambridge New York Port Melbourne New Delhi Singapore: Cambridge University Press, 2024, ISBN: 9781009389433. [Online]. Available: https://d2l.ai/chapter_recurrent-modern/beam-search.html.

[19] Y. Sun, L. Zhu, G. Wang, and F. Zhao, "Multi-Input Convolutional Neural Network for Flower Grading," en, *Journal of Electrical and Computer Engineering*, vol. 2017, e9240407, Aug. 2017, ISSN: 2090-0147. DOI: 10.1155/2017/9240407. [Online]. Available: https://www.hindawi.com/journals/jece/2017/9240407/ (visited on 04/04/2024).

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[21] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385 [cs], Dec. 2015. DOI: 10.48550/arXiv.1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385 (visited on 07/15/2024).

[22] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, arXiv:1711.05101 [cs, math], Jan. 2019. [Online]. Available: http://arxiv.org/abs/1711.05101 (visited on 03/21/2024).

[23] L. Biewald, *Experiment tracking with weights and biases*, Software available from wandb.com, 2020. [Online]. Available: https://www.wandb.com/.

[24] K. Simonyan, A. Vedaldi, and A. Zisserman, *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, en, Dec. 2013. [Online]. Available: https://arxiv.org/abs/1312.6034v2 (visited on 07/14/2024).

[25] P. Liang, R. Bommasani, T. Lee, *et al.*, *Holistic Evaluation of Language Models*, arXiv:2211.09110 [cs], Oct. 2023. [Online]. Available: http://arxiv.org/abs/2211.09110 (visited on 04/29/2024).

[26] Y. Huang, Y. Liu, R. Thirukovalluru, A. Cohan, and B. Dhingra, *Calibrating Long-form Generations from Large Language Models*, arXiv:2402.06544 [cs], Oct. 2024. [Online]. Available: http://arxiv.org/abs/2402.06544 (visited on 11/17/2024).