Analysis of Optimised Transformer Models in Image Captioning Tasks

Maximilian Zimmermann[1,†1,a)] Thang Dang[2] Tsuguchika Tabaru[2] Atsushi Ike[2]

Abstract: This research work is about using Transformer models, which are first introduced in the paper "Attention is All You Need", for a multimodal task, specifically image captioning. By treating it as an NLP translation task, different Transformer models are evaluated and optimised. Through the analysis of the data, model and distributed communication pipeline, bottlenecks are identified and performance increases in regards to accuracy and speed are shown across multiple accelerators.

Keywords: Image captioning, Transformer, Attention

1. Introduction

Image captioning is a research area combining computer vision and natural language processing (NLP), requiring a good understanding of objects, locations and their interaction for image processing. To generate fitting sentences, good syntactic and semantic understanding of language is necessary, making it a challenging task for machine learning algorithms. In contrast to humans, computers are unable to compress dense visual information into language using attention to focus on important information, so traditionally deep neural networks used convolutional architectures to extract and distill features of general objects, but in recent works many modern algorithms based on Transformers [1], [2] are emerging in NLP and computer vision by leveraging the attention mechanism with very promising results. In this paper we incorporate different attention algorithms to the image captioning problem, showcasing their performance in a chemical context under computational restrictions.

In chemistry, one image captioning problem is generating molecular representations, also known as Optical Chemical Structure Recognition (OCSR) [3]. These representations are different ways to indicate information of molecules either in text or image form. It can include the elements, number of atoms and structural bond information of how they are interconnected. In this work we evaluate different architectures on a Kaggle dataset as part of a competition[*1], where the chemical formula is represented as an image, with the goal to generate the International Chemical Identifier (InChI) [4] as text as in **Fig. 1**.

While there are a variety of architectural possibilities, the focus lies on encoder-decoder structures [5] with popular deep-learning techniques like convolution neural networks (CNN) [6], recurrent neural networks (RNN) [7] and Transformers. Especially

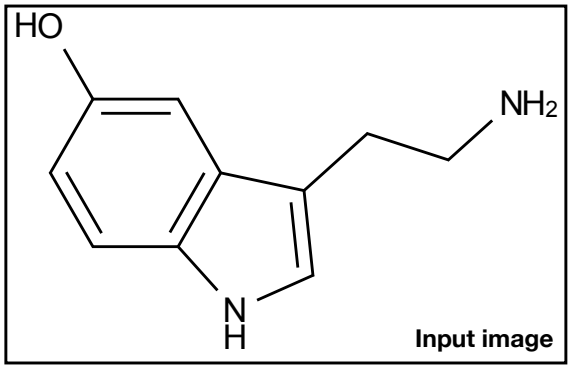

Fig. 1 Example input image (top) and expected generated output (bottom)

self-attention-based architectures, in particular Transformers [1], became a widespread choice in NLP, with many additional research focussed on improving the base architecture. While used extensively for sequence-to-sequence [8] problems like translation, computer vision tasks still primarily use convolutional architectures.

In this work we compare three different architectures; one approach with Bahdanau attention [9] as in [10], one hybrid approach with CNNs and multi-head attention and one purely attention-based Transformer approach shown in **Fig. 2** and further explained in section 4. We show that with each architecture comparable results can be achieved with limited training, but they differ greatly, when comparing preprocessing and training and inference speed.

[1] Hochschule für Technik und Wirtschaft Berlin - University of Applied Sciences
[2] Fujitsu Limited Japan
[†1] Presently with Fujitsu Limited Japan as an intern
[a)] zimmermann_research@mailbox.org
[*1] https://www.kaggle.com/c/bms-molecular-translation/

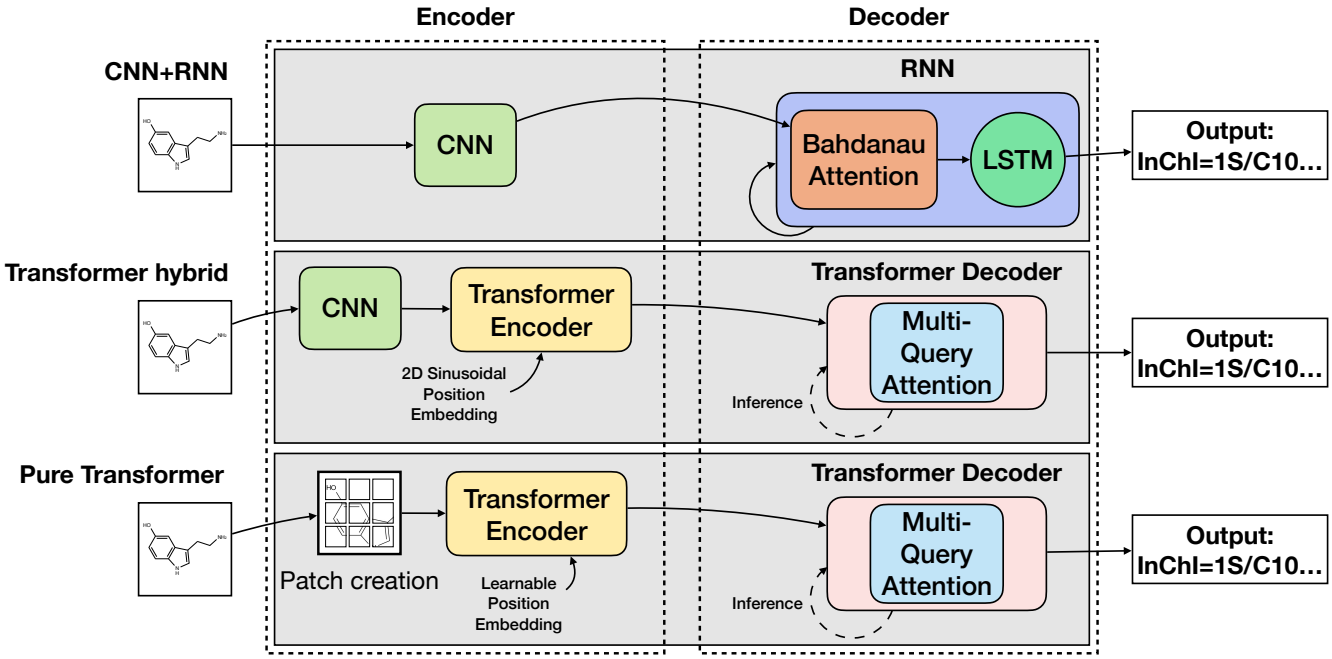

Fig. 2 Model architecture overview

2. Related works

The Show, Attend and Tell model [10] introduced the encoder-decoder architecture from [8] with attention [9] to image captioning and achieved state of the art (SOTA) results in three datasets (Flickr8k, Flickr30k and MS COCO), providing a good baseline algorithm. They apply visual attention in the decoder to specify which extracted features of the image correspond most to each generated word. By visualising how the learned attention connects words to part of images, they demonstrate that visual attention corresponds very well to human intuition.

Following the success of the Transformer by [1] in NLP, the encoder-decoder architecture [5] and multi-head attention [1] became increasingly popular, showing SOTA results in various sequence-to-sequence tasks like machine translation, question answering or text summarisation. Similar to visual attention in images, multi-head attention helps in focussing on relevant parts of input sequences and capturing long-term dependencies in text [11]. Additionally, instead of using a sequential training regime like RNNs, Transformers are able to achieve significant parallelisation, resulting in faster training speeds for shorter sequence lengths. To cope with the limitations of complexity $O(N^2)$ for large sequences, multiple optimised models were proposed [12], introducing solutions like viewing attention through kernelization [13], [14] or limiting attention to predefined patterns [15], [16] with the goal of reducing memory and computational complexity.

With the success in NLP came an increasing interest of Transformer models applied to computer vision tasks, like image processing [17], image recognition [18] or object detection [19]. While early approaches applied attention to CNN extracted features [20], the Vision Transformer (ViT) [18] introduced image patching as a mechanism to directly process image information with attention. This showed promising results on a variety of image classification benchmarks e.g. [21], [22], when pre-trained on large datasets.

In chemistry, early on OCSR were implemented with rule-based approaches [23] or combining image segmentation with aforementioned one [24] to identify different bonds and atoms. Although the latter one used machine vision systems, it still requires specific domain expertise on molecular structures and is time consuming to apply. Chemception [25] showed, that deep-learning models are able to match multi-layer perceptron (MLP) deep neural networks trained on engineered features. Only recently have such machine-learning OCSR methods been published, showing promising results and outperforming non deep-learning ones [3]. A model proposed by [26] to predict SMILES [27], a notation for encoding molecular structures, achieved good accuracy on their Indigo dataset, which is similar to the one used in our work in section 5.1.

By starting with aforementioned Show, Attend and Tell model for image captioning, we evaluate different model architectures in constrained computational environments, building on the great success of Transformer architectures to show accuracy and efficiency of different attention mechanisms.

3. Kaggle environment

Kaggle Inc. is an online community for data scientists and machine learning practitioners by Google LLC, with the primary focus on holding competitions and furthering discussions between users. These competitions are most often held by external companies, which provide a task description, the dataset and prize money. To be able to develop without any self-investment, Kaggle Inc. provides all their users access to Jupyter kernels equipped with CPU, single GPU or a single tensor processing unit (TPU) v3 (8 cores), the latter two limited to certain hour limits per week. A competition runs for a fixed time duration, in which participants can submit predictions for the unknown test set to be evaluated

Table 1 Evaluated models with parameters

	Pure Transformer									CNN+RNN			Transformer hybrid		
	ViT small						ViT base			CNN+RNN			CNN+Transformer		
Variant	A	B	C	D	E	F	G	G*	H	I	J	J*	K	K*	L
Pre-trained				X	X	X	X	X	X						X
Denoising	X					X			X	X					
Augmentation			X	X			X	X			X	X	X	X	X
Denoised test set								X				X		X	
Val score	70.88	74.96	71.82	2.51	2.42	2.51	2.20	2.20	2.22	2.46	11.58	11.58	2.83	2.83	2.10
LB score				5.97	6.36	**3.91**	4.55	**3.07**	3.36	**3.32**	9.76	9.92	5.11	**3.05**	4.10

A,B,C : training stopped after 2 epochs due to no improvement
Val score : Levenshtein distance on validation set
LB score : leaderboard, Levenshtein distance on unknown test set

and are then placed into a leaderboard. Discussions and sharing of ideas in text or code form is proactively supported.

4. Model architectures

We evaluated three different model architectures in this work, described in the following sections and shown in Fig. 2. Each model has an encoder, extracting features from the input images, and a decoder, predicting the InChI-formula in an autoregressive way.

4.1 CNN-RNN

The first model has a CNN as the encoder and uses a RNN for decoding. The encoder is the EfficientNetV2-M model [28], which is also used as CNN for all other models. From the image $x \in \mathbb{R}^{C\times H\times W}$, the extracted image features are reshaped to $x \in \mathbb{R}^{S\times F}$, where C denotes the channels and H, W the height and width. S defines the combined sequence length of $H \times W$ and F the extracted features. The decoder first embeds the input tokens and encoder output to the specified dimensions and then applies Bahdanau attention [9]. After going through an LSTM-Cell it outputs the single next token and the new hidden states. This image captioning architecture was first proposed in [10].

4.2 Transformer hybrid

The second models adds the standard Transformer encoder to the CNN and replaces the RNN with the Transformer decoder. To keep the parameter size around the same we substitute the CNN to the smaller EfficientNetV2-S. It follows the procedure of the first model for reshaping the features, but additionally a fixed 2D sinusoidal position encoding is added. This closely follows the original Transformer, adapted for images. As decoder a modified Transformer decoder is used. Because the computational intensive autoregressive decoding limits the inference speed, two changes, explained in detail in section 5.5, are made. First, we add a cache for the previously computed attention scores of tokens. And secondly, multi-query attention as in [29], where key and value matrices *(K, V)* are shared for each attention head. From here on we refer to this as multi-query decoder.

4.3 Pure Transformer

The third model uses attention exclusively and implements the Vision Transformer (ViT) as in [18], by extracting images into patches of size $x \in \mathbb{R}^{N\times(P^2\times C)}$, where N is the number of patches, P the square patch size and C the number of image channels. These patches are then fed into the encoder. The original classification head is removed. The decoder has the same architecture as in the Transformer hybrid.

5. Experiments

We evaluate the different learning capabilities for the three mentioned architectures. To understand and evaluate the influence of augmentation, model size, input and efficiency, we train four models on several different parameters and input data as shown in **Table 1**. The four models are ViT small and ViT base for the pure Transformer architecture, CNN+Transformer for the Transformer hybrid architecture and a CNN+RNN.

5.1 Dataset

The dataset[*2] is provided on Kaggle by Bristol-Myers Squibb [30] and consists of ~4 million images in varying sizes. These are synthetically generated and additionally preprocessed to include noise, missing atom bonds and rotation. Compared to the training set, the test set images include more noise. For efficient data reading the single images structured in folders are converted to either Tensorflow's TFRecords format or tar-POSIX archives for PyTorch. The images get resized and padded to a fixed size ($H\times W$) of 299×458 pixels, the mean plus an added constant calculated from the first few thousand images. Two versions of the dataset are created, one has all the noise removed and one remains as is. As this is an NLP task, where we generate a sequence of tokens, the Levenshtein distance [31] is used as evaluation metric. It describes the number of token edits (insertion, deletion, substitution) to change one sequence string into another. For the tokenization of the InchI string, we remove the version identifier, split all possible letters for elements, keep numbers in tact, e.g. (15 instead of 1 5), and define fixed tokens like */c*. Because of hardware limitations, the InchI length is fixed to 250 tokens and post-padding is applied. 50k elements are then set aside for validation purposes.

5.2 Model variants

We configured each model with hyper-parameters as in **Table 2**. The parameter count for the encoder is based on the smallest EfficientNetV2 model. For the decoder we used the multi-query decoder with four layers and eight heads as baseline, modifying the hidden size as required by the encoder. To see the effect of different model sizes, we implement ViT in two variants; ViT

[*2] https://www.kaggle.com/c/bms-molecular-translation/

Table 2 Details of trained models

Model architecture	Encoder/Decoder	Model	Layers	Hidden size	MLP size	Heads
Pure Transformer	Encoder	ViT base	12	768	3072	12
Pure Transformer	Encoder	ViT small	8	768	3072	8
Pure Transformer	Decoder	Multi-query decoder	4	768	2048	8
Transformer hybrid	Encoder	CNN+Transformer	5	768	2048	8
Transformer hybrid	Decoder	Multi-query decoder	4	768	2048	8
			Attention units	LSTM hidden state dimension		
CNN+RNN	Decoder	RNN with LSTM-Cell	256	1024		

Layers : number of layers
Hidden size : dimension of each layer, also known as model dimension

small and ViT base.

5.3 Training

We train all models using Adam optimizer [32] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the largest possible batch size for each accelerator. We use a linear warmup for ½ epoch, followed by cosine annealing scheduling as in [33]. The maximum learning rate is set to 1×10^{-3} for 8 accelerators and each model is trained for 9 epochs. Although the CNN could be trained with non-square image sizes as described in section 5.1, they get resized to a square size of 384×384 pixels to have the same input size as the available pre-trained Vision Transformers. Depending on the training run, we also do three different augmentation settings, consisting of random rotation in 90° steps, adding salt-and-pepper noise and cropping the pictures with border distances between 5 to 20. The validation set gets evaluated twice per epoch.

5.4 Metrics

To prevent overfitting we use categorical cross-entropy with label smoothing as the loss function with a value of 0.1 for Transformer-based architectures, similar to the original [1]. The loss calculation also masks padding tokens. For validation and evaluation we calculate the Levenshtein distance as the competition only relies on this metric.

5.5 Optimisations

When running inference for the large test set of ∼1.6 million images, we noticed that autoregressive prediction of the Transformer was not fast enough for sequence predictions of up to 250 tokens in a manageable time frame. As we want to stay as close as possible to the original Transformer architecture we apply two small modifications other than decoupling encoder and decoder.

First is the addition of a cache for attention scores of all previous tokens, so that the complexity for inferencing goes down from $O(MN^2 + N^3)$, where $O(MN)$ is for the encoder-decoder attention between input of size M and current output N. The complexity $O(N^2)$ is for the decoder self-attention over the entire current output. This increases to $O(MN^2)$ and $O(N^3)$ for N tokens to be predicted. The resulting cache implementation reduces the complexity to $O(MN + N^2)$ as only the last token attention has to be calculated.

Secondly we replace multi-head attention by multi-query attention [29]. Normally each head is defined as:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

In multi-query attention, instead of projecting K (keys) and V (values) for each head, only a single set of keys and values is used. This reduces memory and computational requirements.

5.6 Hardware

All training took place on either 8×V100 GPUs or one 8-core TPUv2[*3], both with 16-bit precision enabled for training and inferencing.

6. Evaluation

When evaluating the three architectures, we can see that the Levenshtein distance for the validation set is very close for each architecture, differing only in the range of $0.1 - 0.4$. Also noteworthy is the reached score of ∼2 by every model in validation. The big difference between test set and validation set score further suggests that the test set contains harder to predict samples. This can include unseen bond combinations or numbers, but also a different image composition as for example the test set contains more noise, making more generalisable models a priority.

While the ViT base encoder outperforms the smaller version due to having 2× the parameter size, the experiments show, that with a smaller training dataset and shorter training time, the CNN+Transformer can even outperform larger variants. After only ½ epoch scores reach ∼30 with pre-trained ViT and CNN+RNN, while the CNN+Transformer with pre-training catches up shortly after, making them good few-shot learners.

6.1 Parameter comparison

To clarify the effect of external parameters we implement the ViT small with a combination of different settings, showing the result of each. Non pre-trained ViT perform as expected very bad [18], and are not able to achieve any validation score under 70 with comparable training time. We noticed that although the ViT produced InChI strings, with this amount of training data it was unable learn the relevant patterns and overfitted heavily, independent of preprocessing, augmentation or larger model size. In contrast the convolutional inductive bias helped non pre-trained CNN-based architectures to achieve good results, similar to pre-trained variants, after ∼4 epochs. Preprocessing the data to remove noise seems to help immensely for the test set (LB), improving the score by almost 2.5, therefore being the biggest factor other than pre-training. To alleviate the weakness to noise and rotation, we added augmentation as described in section 5.3, but it shows that the augmentation might be too strong for this short training time, improving the score compared to variant E in

[*3] on Google Colab: https://colab.research.google.com/

Table 3 Comparison of trainable model parameters, total training time and inference time

	Parameters	Training	Inference
CNN+RNN	52.2/12.3	23h04min	23min
CNN+Transformer	48.7/23.5	10h43min	1h02min
ViT base	86.1/23.5	23h53min	2h40min
ViT small	48.3/23.5	15h27min	2h30min

Parameters : encoder/decoder, in million
CNN+RNN : uses 8-core TPUv2
Training time : time taken for 9 epochs

Table 4 Optimisation results for inferencing

	Step time	Batch size	Speed-up
Base	33.61s	250	
Cache	12.82s	250	2.62×
MQD	32.32s	250	1.04×
Cache+MQD	10.16s	250	3.31×

Base : original Transformer implementation
MQD : decoder with multi-query attention
Benchmarks done on one V100 GPU

Table 1, but falling behind by a big margin in regards to the denoised variant F. For multi-head attention-based variants, the best scores show the positive effects of augmentation when predicting on a denoised test set. For computer vision tasks like image classification it was shown, that deeper models with more parameters achieve better accuracy [18] and this holds true for image captioning as well, with ViT base outperforming the smaller variant by ∼1.5. Because the encoder only runs once for each prediction batch (section 5.5), the total inference time only increases slightly due to the larger encoder.

6.2 Efficiency

We also note the difference in training and inference time of each architecture in **Table 3**, although they have around the same parameter size. When comparing total training time for nine epochs, the CNN+Transformer is the fastest architecture for training and achieves good results thanks to the CNN feature extraction. Interestingly, even though the ViT base is larger, it takes around the same time as the CNN+RNN to train and produces better results. While the CNN+RNN has the shortest decoding time for inferencing, the CNN+Transformer is the fastest Transformer-based architecture. Additionally it was shown in [34], that for CNN and RNN based architectures, TPUs are even faster than a comparable setup of GPUs.

To compare the effect of our inference optimisations, we profile each version and note the speed-ups in terms of step time with the same architecture and batch size in **Table 4**. Especially caching provides a big speed-up without any impact on model performance. Multi-query attention on its own doesn't provide much speed-up, problem being either the relatively small decoder or too small of a batch size as the authors noted [29]. It has a larger effect when combined with caching, reaching a total speed-up of ∼3.3×, cutting inference time to just over 1h on 8 GPUs for the fastest model.

7. Conclusion

In this work we presented an analysis of different deep-learning architectures with attention on an image captioning task in a chemical context. We showed that with restricted computing power very good results on OCSR can be achieved with little to no domain knowledge, showing promise for deep-learning architectures in other areas as well. The combination of CNNs and Transformers allows for very fast and efficient training compared to the CNN+RNN by leveraging the parallelizable architecture of Transformers and inductive bias of CNNs. Although attention models are able to achieve good results on their own, our experiments show, that additional processing for images, specifically denoising with augmentations, can be very beneficial. With small modifications the Transformer can be speed-up even for inference-heavy applications, making it very versatile all around.

Acknowledgments This research was funded by the "Vulcanus in Japan" programme[*4] and Fujitsu Limited Japan.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I.: Attention Is All You Need, *arXiv:1706.03762 [cs]* (2017).
[2] Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H. and Douze, M.: LeViT: A Vision Transformer in ConvNet's Clothing for Faster Inference, *arXiv:2104.01136 [cs]* (2021).
[3] Rajan, K., Brinkhaus, H., Steinbeck, C. and Zielesny, A.: A Review of Optical Chemical Structure Recognition Tools, *Journal of Cheminformatics*, Vol. 12 (online), DOI: 10.1186/s13321-020-00465-0 (2020).
[4] Heller, S. R., McNaught, A., Pletnev, I., Stein, S. and Tchekhovskoi, D.: InChI, the IUPAC International Chemical Identifier, *Journal of Cheminformatics*, Vol. 7, No. 1, p. 23 (online), DOI: 10.1186/s13321-015-0068-4 (2015).
[5] Cho, K., van Merrienboer, B., Bahdanau, D. and Bengio, Y.: On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, *arXiv:1409.1259 [cs, stat]* (2014).
[6] Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J.: A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21 (online), DOI: 10.1109/TNNLS.2021.3084827 (2021).
[7] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J. and Khudanpur, S.: Recurrent Neural Network Based Language Model, Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010, Vol. 2, pp. 1045–1048 (2010).
[8] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *arXiv:1409.3215 [cs]* (2014).
[9] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *arXiv:1409.0473 [cs, stat]* (2016).
[10] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. and Bengio, Y.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, *arXiv:1502.03044 [cs]* (2016).
[11] Chaudhari, S., Mithal, V., Polatkan, G. and Ramanath, R.: An Attentive Survey of Attention Models, *arXiv:1904.02874 [cs, stat]* (2020).
[12] Tay, Y., Dehghani, M., Bahri, D. and Metzler, D.: Efficient Transformers: A Survey, *arXiv:2009.06732 [cs]* (2020).
[13] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L. and Weller, A.: Rethinking Attention with Performers, *arXiv:2009.14794 [cs, stat]* (2021).
[14] Katharopoulos, A., Vyas, A., Pappas, N. and Fleuret, F.: Transformers Are RNNs: Fast Autoregressive Transformers with Linear Attention, *arXiv:2006.16236 [cs, stat]* (2020).
[15] Qiu, J., Ma, H., Levy, O., Yih, S. W.-t., Wang, S. and Tang, J.: Blockwise Self-Attention for Long Document Understanding, *arXiv:1911.02972 [cs]* (2020).
[16] Child, R., Gray, S., Radford, A. and Sutskever, I.: Generating Long Sequences with Sparse Transformers, *arXiv:1904.10509 [cs, stat]* (2019).
[17] Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C. and Gao, W.: Pre-Trained Image Processing Transformer, *arXiv:2012.00364 [cs]* (2021).
[18] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N.: An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale, *arXiv:2010.11929 [cs]*

[*4] https://www.eu-japan.eu/events/vulcanus-japan

(2021).
[19] Zhu, X., Su, W., Lu, L., Li, B., Wang, X. and Dai, J.: Deformable DETR: Deformable Transformers for End-to-End Object Detection, *arXiv:2010.04159 [cs]* (2021).
[20] Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A. and Tran, D.: Image Transformer, *arXiv:1802.05751 [cs]* (2018).
[21] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L.: Imagenet: A Large-Scale Hierarchical Image Database, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Ieee, pp. 248–255 (2009).
[22] Krizhevsky, A., Nair, V. and Hinton, G.: CIFAR-10 (Canadian Institute for Advanced Research).
[23] Sadawi, N. M., Sexton, A. and Sorge, V.: Chemical Structure Recognition: A Rule-Based Approach, *Electronic Imaging* (2011).
[24] Park, J., Rosania, G. R., Shedden, K. A., Nguyen, M., Lyu, N. and Saitou, K.: Automated Extraction of Chemical Structure Information from Digital Raster Images., *Chemistry Central journal*, Vol. 3, p. 4 (online), DOI: 10.1186/1752-153X-3-4 (2009).
[25] Goh, G. B., Siegel, C., Vishnu, A., Hodas, N. O. and Baker, N.: Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-Developed QSAR/QSPR Models, *arXiv:1706.06689 [cs, stat]* (2017).
[26] Staker, J., Marshall, K., Abel, R. and McQuaw, C.: Molecular Structure Extraction From Documents Using Deep Learning, *arXiv:1802.04903 [physics]* (2018).
[27] Weininger, D.: SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules, *Journal of Chemical Information and Computer Sciences*, Vol. 28, No. 1, pp. 31–36 (online), DOI: 10.1021/ci00057a005 (1988).
[28] Tan, M. and Le, Q. V.: EfficientNetV2: Smaller Models and Faster Training, *arXiv:2104.00298 [cs]* (2021).
[29] Shazeer, N.: Fast Transformer Decoding: One Write-Head Is All You Need, *arXiv:1911.02150 [cs]* (2019).
[30] Bristol-Myers Squibb: Bristol-Myers Squibb – Molecular Translation (2021).
[31] Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals, *Soviet physics. Doklady*, Vol. 10, pp. 707–710 (1965).
[32] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *arXiv:1412.6980 [cs]* (2017).
[33] Loshchilov, I. and Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts, *arXiv:1608.03983 [cs, math]* (2017).
[34] Wang, Y. E., Wei, G.-Y. and Brooks, D.: Benchmarking TPU, GPU, and CPU Platforms for Deep Learning, *arXiv:1907.10701 [cs, stat]* (2019).